สำนักหอสมุดกลาง_พระจอมเกล้าลาดกระบัง

เครื่องกำเนิดสัญญาณทดสอบจอมอนิเตอร์โดยFPGA

FPGA COLOR MONITOR TESTER

โดย

นาย บุญเลิศ วรรณพงษ์

นาย ศรเพชร ทองนำ

วพ.
2156ค
2547

เลขหมู่.....
เลขทะเบียน...61334...
วัน,เดือน,ปี 17 ก.ค. 2549

b.11568306
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

FPGA COLOR MONITOR TESTER

BY

BOONLERT WANNAPONG

SORNPETCH THONGNAM

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

THE REQUIREMENT FOR THE DEGREE

BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2004

หัวข้อปริญญานิพนธ์	เครื่องกำเนิดสัญญาณทดสอบจอมอนิเตอร์โดยFPGA
ชื่อนักศึกษา	1. นาย บุญเลิศ วรรณพงษ์ รหัสประจำตัว 45015800
	2. นาย ศรเพชร ทองนำ รหัสประจำตัว 45015818
อาจารย์ที่ปรึกษา	ผศ.อุทัย ศรีธีระวิโรจน์
ระดับการศึกษา	ปริญญาตรี วิศวกรรมศาสตรบัณฑิต
	สาขาวิศวกรรมสารสนเทศ
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2547

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้เอกสารฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

..... ประธานกรรมการ
( )
..... กรรมการ
( )
..... กรรมการ
( )
..... กรรมการ
( )
..... กรรมการ
( )

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อปริญญานิพนธ์	เครื่องกำเนิดสัญญาณทดสอบจอมอนิเตอร์โดยFPGA
ชื่อนักศึกษา	นาย บุญเลิศ วรรณพงษ์ รหัสประจำตัว 45015800
	นาย ศรเพชร ทองนำ รหัสประจำตัว 45015818
อาจารย์ที่ปรึกษา	ผศ.อุทัย ศรีธีระวิโรจน์
ระดับการศึกษา	ปริญญาตรี วิศวกรรมศาสตรบัณฑิต
	สาขาวิศวกรรมสารสนเทศ
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2547

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการสร้างเครื่องทดสอบสัญญาณจอมอนิเตอร์โดยไม่ต้องเชื่อมต่อกับเครื่องคอมพิวเตอร์ โดยใช้ FPGA เพื่อลดความซับซ้อนของวงจรประเภทอุปกรณ์ทางลอจิก เพิ่มความสะดวกในการสร้างเพื่อนำไปใช้งานและการนำไปพัฒนา โดยมีส่วนประกอบหลักคือชุดสร้างความถี่มาตรฐานและส่วนของชุดควบคุมการแสดงผลที่จอ ซึ่งนำอุปกรณ์FPGA มาเป็นส่วนควบคุมการทำงาน การเลือกฟังก์ชัน คาบเวลาในการแสดงผล เพื่อให้ได้ลักษณะของการทดสอบที่แตกต่างกัน โดยใช้ภาษา VHDL ในการเขียนโปรแกรมให้กับ FPGA เพราะเข้าใจได้ง่ายและ นำไปประยุกต์ใช้กับงานอื่นๆได้ด้วย

Thesis Title	FPGA Color Monitor Tester
Student	BOONLERT WANNAPONG ID 45015800
	SORNPETCH THONGNAM ID 45015818
Advisor	Asst. Prof. U-thai Sritheeravirojana
Graduate Level	Bachelor Degree of Information Engineering
Departmaent	Information Engineerring
Acadamic Year	2004

ABSTRACT

This thesis is present FPGA Color Monitor Tester that can work without computer by using FPGA to reduce difficult of logic circuit and easier to further development. Consist of 2 main part : Fundamental Frequency Generator and Display Controller which is using FPGA to control a Display Part, choose function and period to display and it use , choose function, period for display and use VHDL to program FPGA because it easy to understand it language and can apply to otherwork.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี ด้วยความช่วยเหลือและชี้แนะจากหลายท่าน ผู้จัดทำขอขอบพระคุณอาจารย์ที่ปรึกษา ผศ. อุทัย ศรีธีระวิโรจน์ ที่กรุณาให้คำปรึกษาและช่วยเหลือด้านข้อมูลตลอดทั้งอุปกรณ์ในการทำปริญญานิพนธ์มาโดยตลอด และให้ความเมตตาต่อศิษย์ด้วยดีเสมอมา ขอขอบคุณเพื่อนนักศึกษาปริญญาตรี และเพื่อนนักศึกษาภาควิชาวิศวกรรมโทรคมนาคมที่ให้คำแนะนำ และเอื้อเฟื้ออุปกรณ์ในการทดลอง

สุดท้ายขอกราบขอบพระคุณ บิดา มารดา และพี่ๆ ที่ให้ความสำคัญกับการศึกษาและให้กำลังใจด้วยดีตลอดมา

นายบุญเลิศ วรรณพงษ์

นายศรเพชร ทองนำ

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูป	ฉ
สารบัญตาราง	ช
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 จุดประสงค์	1
1.3 ขอบเขตของโครงงาน	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 หลักการทำงานของจอมอนิเตอร์และเอฟพีจีเอ	
2.1โครงสร้างของจอมอนิเตอร์	3
2.2 FPGA	13
2.3 การออกแบบทางด้านซอร์ฟแวร์โดยภาษาVHDL	20
บทที่ 3 การออกแบบ	38
3.1 วงจรผลิตความถี่และสัญญาณซิงค์	39
3.2 การออกแบบวงจรเอาต์พุต	41
3.3 ลักษณะชิ้นงาน	43
บทที่ 4 ผลการทดลอง	
4.1 วัตถุประสงค์การทดลอง	45
4.2 อุปกรณ์การทดลอง	42
4.3 การทดลองใช้งาน	42
4.4 วิเคราะห์ผลการทดลอง	54
บทที่ 5 สรุปผลการทดลองและปัญหาที่พบ	
5.1 สรุปผลการทดลอง	56
5.2 ปัญหาที่พบ	56

สารบัญ(ต่อ)

หน้าที่

5.3 แนวทางในการแก้ปัญหา 56

บรรณานุกรม 57

ภาคผนวก

ภาคผนวก ก. บอร์ดทดลอง PLD – A01

ภาคผนวก ข. โปรแกรมภาษาVHDL และวงจรที่ใช้งาน

ภาคผนวก ค. ข้อมูล (Data Sheet)

ภาคผนวก ง. คู่มือการใช้งาน

สารบัญรูป

รูปที่ หน้า

รูปที่ 2.1 D –Sub Connector 3
รูปที่ 2.2 ผังเบื้องต้นของภาคจ่ายแรงดันไฟสูงเพื่อจ่ายให้กับมอนิเตอร์ 5
รูปที่ 2.3 แผนผังแสดงการทำงานของภาคฮอริซอนตอล 6
รูปที่ 2.4 แผนผังเบื้องต้นของภาคเวอร์ติคอล 7
รูปที่ 2.5 ความถี่ที่ผลิตโดยวงจรเวอร์ติคอลออสซิลเลเตอร์ 8
รูปที่ 2.6 แผนผังเบื้องต้อนของวงจรขยายสัญญาณภาพ 8
รูปที่ 2.7 แผนผังเบื้องต้นของหลอดภาพ 9
รูปที่ 2.8 แผนผังการควบคุมและไมโครโปรเซสเซอร์ของมอนิเตอร์ 10
รูปที่ 2.9 แผนผังของไมโครโปรเซสเซอร์และวงจรสนับสนุน 11
รูปที่ 2.10 แผนผังของลิเนียร์เรกูเลเตอร์ 12
รูปที่ 2.11 แผนผังของสวิตซิ่งงโหมดเพาเวอร์ซัพพลาย 12
รูปที่ 2.12 โครงสร้างภายในของ FPGA ตระกูล MAX7000S 14
รูปที่ 2.13 โครงสร้างภายในของ FPGA ตระกูล FLEX10K 15
รูปที่ 2.14 การโปรแกรมลงในชิพ 16
รูปที่ 2.15 ขั้นตอนการออกแบบระบบดิจิตอล 20
รูปที่ 2.16 การออกแบบระบบเส้นทางของข้อมูล 21
รูปที่ 2.17 การกำหนดการเชื่อมต่อและสถาปัตยกรรม 25
รูปที่ 2.18 บล็อคไดอะแกรมและการบรรยายการเชื่อมต่อของ clock_component 26
รูปที่ 2.19 การบรรยายเชิงพฤติกรรมของ clock_component 27
รูปที่ 2.20 โครงสร้างทั่วไปของส่วนการประกาศแพ็กเกจ 28
รูปที่ 2.21 โครงสร้างของบอดี้แพ็กเกจ 28
รูปที่ 2.22 โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ 29
รูปที่ 2.23 การใช้โพรซีเจอร์ 29
รูปที่ 2.24 การใช้ฟังก์ชั่น 29
รูปที่ 2.25 ตัวดำเนินการใน VHDL 30
รูปที่ 2.26 รูปแบบของการบรรยายแบบโปรเซส 32
รูปที่ 2.27 ตัวอย่างการประกาศตัวดำเนินการภายในโปรเซส 32

สารบัญรูปภาพ(ต่อ)

รูปที่ 2.28 เงื่อนไขการกระทำในโปรเซส 33

รูปที่ 2.29 แสดงการกระทำในโปรเซส 33

รูปที่ 2.30 (a) ตัวอย่างโมเดล D-Flip Flop, (b) การบรรยายการเชื่อมต่อของ D-Flip Flop 34

รูปที่ 2.31 การบรรยายเชิงพฤติกรรมของ D-Flip Flop 35

รูปที่ 2.32 ขั้นตอนการออกแบบจากบนลงล่าง 36

รูปที่ 3.1 Horizontal Timing 40

รูปที่ 3.2 Vertical Timing 40

รูปที่ 3.3 สัญญาณ สีแดง สีเขียว สีน้ำเงิน กรณีแสดงสีขาว 41

รูปที่ 3.4 แสดงการติดต่อของ FPGA กับสัญญาณต่างๆ 42

รูปที่ 3.5 แสดงการต่อพอร์ทจาก FPGA ไปยัง DB15 เพื่อแสดงผลที่จอมอนิเตอร์ 42

รูปที่ 3.6 แสดงด้านบนของชิ้นงาน 43

รูปที่ 3.7 แสดงด้านหน้าของชิ้นงาน 44

รูปที่ 3.8 แสดงด้านหลังของชิ้นงาน 44

รูปที่ 4.1 แสดงบอร์ด FPGA PLD A01(EPM7128SLC84-15) 45

รูปที่ 4.2 แสดงด้านหน้าของชิ้นงาน สวิทช์เลือกฟังก์ชัน 46

รูปที่ 4.3 แสดงด้านหลังของชิ้นงาน ช่องต่อใช้งาน และช่องต่อแหล่งจ่ายไฟ 46

รูปที่ 4.4 แสดงจุดวัดสัญญาณ 47

รูปที่ 4.5 แสดงผลการทำงานฟังก์ชันสีขาว 47

รูปที่ 4.6 รูปสัญญาณสีแดง 48

รูปที่ 4.7 รูปสัญญาณสีเขียว 48

รูปที่ 4.8 รูปสัญญาณสีน้ำเงิน 49

รูปที่ 4.9 รูปสัญญาณHorizontal sync 49

รูปที่ 4.10 รูปสัญญาณVertical sync 50

รูปที่ 4.11 แสดงผลการทำงานฟังก์ชันสีแดง 50

รูปที่ 4.12 รูปสัญญาณสีเขียวเทียบสีแดง 51

รูปที่ 4.13 รูปสัญญาณสีเขียวเทียบสีน้ำเงิน 51

สารบัญรูปภาพ(ต่อ)

รูปที่ 4.14 แสดงผลการทำงานฟังก์ชันแถบสี 52
รูปที่ 4.15 รูปสัญญาณสีแดงเทียบสีเขียว 52
รูปที่ 4.16 รูปสัญญาณสีแดงเทียบสีน้ำเงิน 52
รูปที่ 4.17 แสดงผลการทำงานฟังก์ชันอื่นๆ 54

สารบัญตาราง

ตารางที่ หน้า

ตารางที่2.1 แสดงหน้าที่ของขาแต่ละขาของคอนเน็คเตอร์ดีซับ 3

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

โดยปกติถ้าเราจะทดสอบการทำงานของจอมอนิเตอร์ เราจะต้องทำการนำเครื่องคอมพิวเตอร์มาต่อกับจอมอนิเตอร์ แล้วใช้โปรแกรมสร้างสัญญาณทดสอบจอมอนิเตอร์ทำให้ไม่สะดวกต่อการใช้งานและจะมีปัญหามากขึ้นเมื่อต้องการทดสอบจอมอนิเตอร์ในจำนวนมากๆจึงได้หาวิธีการที่ทำให้การทดสอบจอมอนิเตอร์ทำได้สะดวกมากขึ้น

ปัจจุบันมีเทคโนโลยีต่างๆให้เลือกใช้มากมาย การพัฒนาอุปกรณ์ต่างๆทำได้ง่ายขึ้น ขีดความสามารถการทำงานของอุปกรณ์เพิ่มสูงขึ้น FPGAก็เป็นส่วนหนึ่งที่ได้มีการพัฒนาขึ้น โดยมีจุดเด่นคือ มีความสามารถโปรแกรมวงจรทางลอจิกเข้าไปยังFPGA เพื่อนำไปใช้งานได้โดยตรง จึงทำให้เหมาะสมในการนำมาใช้โครงงานนี้ เนื่องจากโครงงานนี้มีส่วนประกอบหลักเป็นวงจรทางลอจิก ถ้าเราใช้เป็นตัวอุปกรณ์อย่างเช่น IC จะเห็นได้ว่าต้องใช้IC ในปริมาณที่มากไม่มีความสะดวกและมีความซับซ้อนของวงจร ดังนั้นจึงได้นำอุปกรณ์FPGA นำมาประยุกต์ใช้ทำวงจรทางลอจิกช่วยลดอุปกรณ์ที่ใช้ทำวงจร และความซับซ้อนของวงจรลงมีความสะดวกในการนำไปออกแบบมากขึ้น

1.2 จุดประสงค์

1.2.1 ต้องการพัฒนาเครื่องมือทดสอบและตรวจสอบจอมอนิเตอร์สำหรับให้ความสะดวกในการใช้งานของช่างเทคนิคในการให้บริการ

1.2.2 นำอุปกรณ์ที่สามารถจัดซื้อได้ภายในประเทศนำมาใช้งานได้

1.2.3 เพื่อศึกษาการใช้งาน FPGA

1.2.4 เพื่อศึกษาการทำงานของวงจรดิจิตอล และอุปกรณ์อิเล็กทรอนิกส์

1.2.5 เพื่อศึกษาเกี่ยวกับทฤษฎีของแสง สีของจอมอนิเตอร์

1.3 ขอบเขตของโครงงาน

1.3.1 สามารถส่งสัญญาณทดสอบไปยังจอมอนิเตอร์

1.3.2 สามารถประยุกต์ใช้สำหรับการตรวจสอบจอมอนิเตอร์เพื่อใช้ซ่อมจอมอนิเตอร์ได้

1.4 ผลคาดว่าที่จะได้รับ

1.4.1 สามารถนำความรู้ของทำงานจอมอนิเตอร์ไปประยุกต์ใช้ได้

1.4.2 สามารถประยุกต์ใช้อุปกรณ์ FPGA ในการสร้างวงจรเพื่อใช้งานได้

1.4.3 สามารถที่จะนำทฤษฎีที่ได้จากการเรียนรู้มาประยุกต์ใช้ในการทำงานในทางปฏิบัติได้

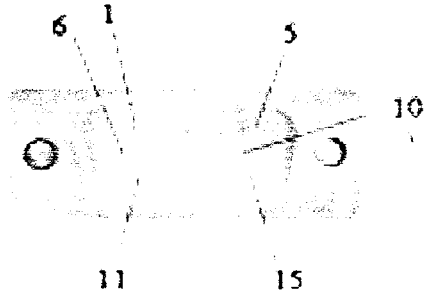
บทที่ 2
หลักการทำงานของจอมอนิเตอร์และเอฟพีจีเอ

2.1โครงสร้างของจอมอนิเตอร์

มอนิเตอร์แบบมัลติซิงค์

ระบบมัลติซิงค์หรือซิงค์มาสเตอร์เป็นการนำเอาเทคโนโลยีของการทวีคูณความถี่(Multiple frequency technology) มาใช้ มอนิเตอร์ประเภทนี้เป็นประเภทที่สามารถปรับความถี่ได้อัตโนมัติเมื่อนำเข้าไปต่อเข้ากับการ์ดของระบบภาพในการสแกนความถี่แบบต่างๆตามที่ค่ารีโซลูชั่น (Resolution : ความละเอียดของสัญญาณภาพ)ที่เครื่องคอมพิวเตอร์ต้องการ มอนิเตอร์ในรูปแบบของมัลติซิงค์(Multisync) หรือซิงค์มาสเตอร์(Sync Master) จึงถูกออกแบบให้มีการทวีคูณความถี่ให้รองรับกับคำสั่งของสัญญาณภาพที่มาจากการ์ดของเครื่องคอมพิวเตอร์

โครงสร้างการส่งข้อมูลของวงจรมอนิเตอร์



รูป 2.1 D-Sub Connector

ตาราง 2.1 แสดงหน้าที่ของขาแต่ละขาของคอนเน็คเตอร์ดีซับ

ลำดับที่	ชนิดของซิงค์		
	Separate	**Composite**	**Sync on Green**
1	Red	Red	Red
2	Green	Green	Green + Sync
3	Blue	Blue	Blue
4	GND	GND	GND
5	DDC Return	DDC Return	DDC Return

6	GND-R	GND-R	GND-R
7	GND-G	GND-G	GND-G
8	GND-B	GND-B	GND-B
9	Reserved	Reserved	Reserved
10	GND-Sync/Self-Raster	GND-Sync/Self-Raster	GND-Sync/Self-Raster
11	GND	GND	GND
12	DDC Data	DDC Data	DDC Data
13	H-Sync	H/V-Sync	Not Used
14	V-Sync	Not Used	Not Used
15	DDC Clock	DDC Clock	DDC Clock

ระบบไบอัสหลอดภาพ

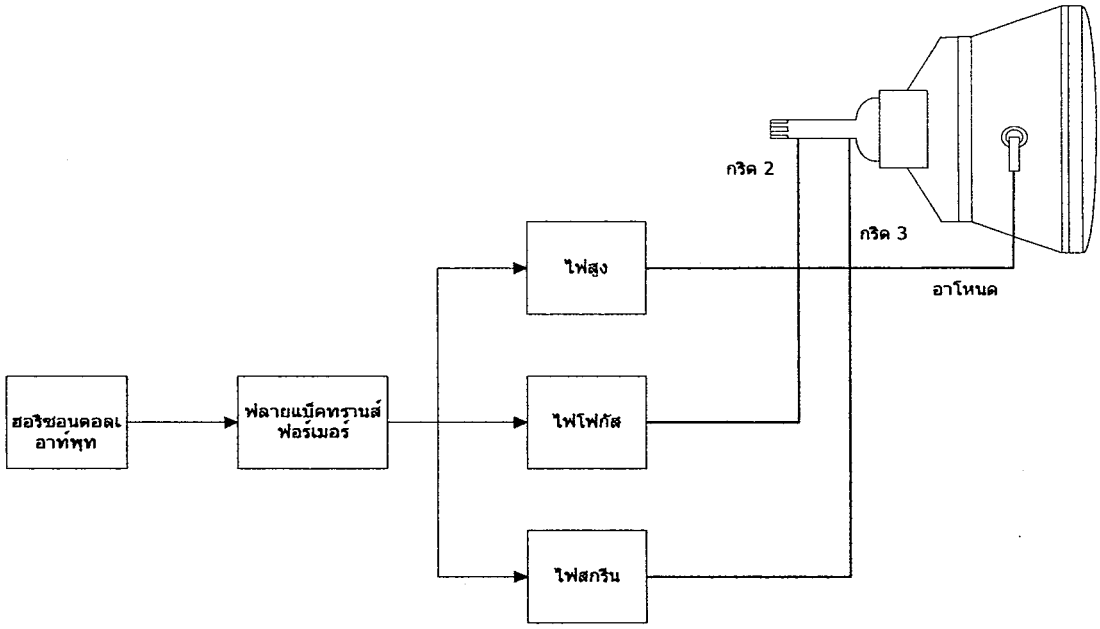
การที่จอมอนิเตอร์จะทำงานได้ วงจรหลอดภาพจะต้องมีแรงดันไฟจ่ายเป็นไบอัสให้ก่อน โดยที่ภาคจ่ายไฟต้องทำงานก่อน หลังจากนั้นระบบแรงดันไฟสูงต้องทำงานในลำดับต่อมา

ภาคกำเนิดแรงดันไฟสูงเป็นภาคการทำงานที่ต่อเนื่องจากภาคฮอริซอนตอล โดยหลักการแล้วภาคฮอริซอนตอลหรือภาคผลิตความถี่ไลน์ ทำหน้าที่บังคับเส้นแสงของลำอิเล็กตรอนที่ยิงไปยังจอให้กวาดลำแสงกำเนิดเป็นเส้นภาพขึ้นมากับเอาประโยชน์มาใช้เพื่อกำเนิดแรงดันไฟสูง

จอคอมพิวเตอร์ก็คือจอภาพแบบเดียวกับจอทีวี ซึ่งพัฒนามาจากจอCRTแบบเดิม โดยจอมอนิเตอร์ต้องการแรงดันในช่วง 20,000-30,000 โวลต์ (ขึ้นอยู่กับขนาดของจอ) เพื่อจ่ายให้กับอาโนดของจอ แรงดันไฟสูงที่จ่ายไปยังอาโนดก็เพื่อเป็นไฟสูงล่อประจุลำอิเล็กตรอนจากคาโถดที่ทำหน้าที่เป็นกระบอกปืนอิเล็กตรอน สามารถดึงอิเล็กตรอนให้วิ่งไปที่หน้าจอให้ได้ หากแรงดันไฟต่ำแรงดึงดูดของอิเล็กตรอนจะไม่มีแรงดึงเพียงพอ

เมื่ออิเล็กตรอนเป็นประจุไฟฟ้าศักย์ลบแรงดันไฟสูงที่อาโนดเป็นแรงดันศักย์บวก จากหลักการของประจุไฟฟ้า เมื่อประจุศักย์ต่างกันจะวิ่งเข้าหากัน เมื่อเราสร้างปืนอิเล็กตรอนให้เป็นตัวบีบลำแสงเพื่อส่งอิเล็กตรอนให้วิ่งและแรงดันไฟสูงที่อาโนดหรือที่หน้าจอมีแรงดึงดูดสูง จึงทำให้อิเล็กตรอนวิ่งตรงไปยังหน้าจอได้ หน้าจอได้เคลือบสารเรืองแสงเอาไว้เมื่ออิเล็กตรอนถูกออกแบบให้ส่งออกด้วยการบีบทิศทาง จึงทำให้แทนที่จะวิ่งเข้าหาอาโนด แต่วิ่งไปยังหน้าจอแทน เกิดแสงสว่างพุ่งชนสารเรืองแสง

การกำเนิดแรงดันไฟสูงมีอุปกรณ์หลักคืออินทิเกรทไฮโวลเตจทรานส์ฟอร์เมอร์หรือ IHVT หรือเรียกกันสั้นๆว่า ฟลายแบ็คทรานฟอร์เมอร์



รูป 2.2 ผังเบื้องต้นของภาคจ่ายแรงดันไฟสูงเพื่อจ่ายให้กับมอนิเตอร์

จากรูปเป็นแผนผังเบื้องต้นของวงจรภาคผลิตแรงดันไฟสูงของมอนิเตอร์สำหรับคอมพิวเตอร์ ซึ่งมีอุปกรณ์สำคัญ 2 ส่วน คือ จอ CRT และฟลายแบ็คทรานส์ฟอร์เมอร์ โดยหลักการของการผลิตแรงดันไฟสูงอาศัยหลักการคอนเวอร์เตอร์ โดยภาคฮอริซอนตอลที่เป็นวงจรความถี่สูงผลิตความถี่ขึ้น เมื่อความถี่สูงส่งขับกระแสออกไปยังฟลายแบ็คทรานฟอร์เมอร์ด้วยหลักการสะสมพลังงาน (Induce) จึงทำให้แรงดันไฟสูงขึ้น เพราะเหตุผลของระบบสะสมพลังงานผ่านสนามแม่เหล็กของขดลวด ยิ่งความถี่สูงขึ้น ยิ่งสะสมพลังงานได้มาก จึงทำให้คอนเวอร์เตอร์กำเนิดแรงดันไฟสูงขึ้นได้

แรงดันไฟสูงที่เกิดขึ้นยังถูกนำไปแบ่งเป็นไฟไบอัสหลอดภาพอีกทางหนึ่งโดยอุปกรณ์ภายในหลอดภาพที่ต้องการแรงดันไฟสูงได้แก่ กริดอันดับ2 (G2) ซึ่งเรียกว่า “สกรีนกริด” และกริดอันดับ3 ซึ่งเรียกว่า “โฟกัสกริด” ซึ่งG2ต้องการแรงดันไฟ 400-500 โวลต์ และ G3 ต้องการแรงดันไฟประมาณ 5,000โวลต์

สกรีนกริดทำหน้าที่เร่งลำอิเล็กตรอนจากปืนอิเล็กตรอนให้วิ่งไปที่จอภาพอย่างมีประสิทธิภาพ ถ้าแรงดันไฟต่ำไปย่อมทำให้จอมัวลง

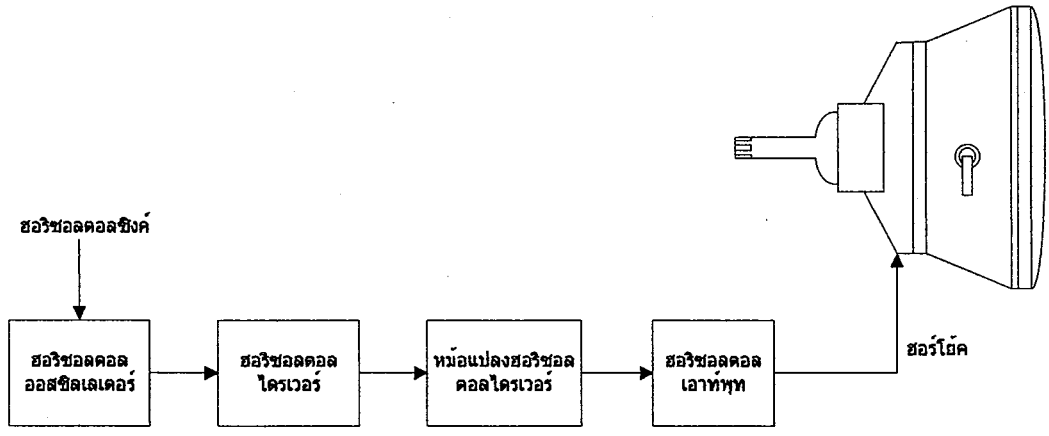
ในขณะที่โฟกัสกริด ทำหน้าที่บีบลำอิเล็กตรอนให้เกิดลำแสงคมชัดตรงจุดสี แรงดันไฟโฟกัสต้องอาศัยกระบวนการปรับแต่งเพื่อให้ภาพมีความคมชัด หากไฟส่วนนี้บกพร่องย่อมทำให้ภาพที่ออกมาขาดความคมชัด

ในส่วนของภาคฮอริซอนตอลเองรับแรงดันไฟต้นทาง 100-200 โวลต์เท่านั้น ซึ่งเป็นวิธีการเดียวกับเครื่องรับโทรทัศน์ การจ่ายแรงดันไฟที่สูงมากไปย่อมทำให้วงจรมีอายุการใช้งานสั้นลง ดังนั้นภาคฮอริซอนตอล, ภาคจ่ายไฟและวงจรหลอดภาพจึงมักเป็นภาคที่เกิดความบกพร่องเสียหายมากกว่าภาคอื่นๆ

ระบบควบคุมการสแกนแนวนอน หรือภาคฮอริซอนตอล

การทำให้เกิดภาพขึ้นที่จอต้องอาศัยการกวาดลำแสงที่มีการบังคับแสงมากน้อยเพื่อกำเนิดภาพ แต่ภาพย่อมมีพื้นที่มีเฟรมเหมือนกับการที่เราเห็นภาพ 1 ภาพ ใน 1 เฟรมหรือ 1 พื้นที่ของการเห็นภาพมีวิวใดๆบ้าง การกำเนิดภาพของจอจึงต้องมีการกำหนดเฟรมภาพที่เกิดกับจอ วงจรที่กำเนิดเฟรมภาพต้องสร้างเฟรมขึ้นมา จอโทรทัศน์และจอมอนิเตอร์จึงต้องมีการบังคับ เพื่อกวาดลำอิเล็กตรอนทั้งแนวซ้าย-ขวา และ บน-ล่าง

สำหรับการกำหนดภาพให้เกิดการกวาดหรือเขียนภาพในแนวนอนหรือแนวการกวาดซ้าย-ขวา มีความสำคัญต่อการทำให้เกิดภาพหรือเส้นภาพ ถูกเรียกตามศัพท์ภาษาเยอรมันว่า “ฮอริซอนตอล (Horizontal)” แต่หากใช้ภาษาอังกฤษจะใช้คำว่าการทำให้เกิดเป็นเส้นภาพหรือ “ไลน์ (Line)” การทำงานของระบบนี้ หากพิจารณาจากรูปที่2.3 ภาคฮอริซอนตอลต้องทำหน้าที่บังคับลำอิเล็กตอนให้มีการเบี่ยงเบนหรือเส้นภาพในแนวนอน โดยการบังคับผ่านวงจรขดลวดที่พันและวางไว้อยู่บริเวณคอหลอดภาพ ขดลวดเพื่อทำให้ลำอิเล็กตรอนเบี่ยงเบนเรียกเป็นภาษาเยอรมันว่า “โย้ค” หรือ “ฮอริซอนตอล ดีเฟลกชั่น โย้ค ” (Horizontal deflection yoke)



รูปที่ 2.3 แผนผังแสดงการทำงานของภาคฮอริซอนตอล

การกวาดลำแสงหรือการเบี่ยงเบนลำอิเล็กตรอนใช้คำว่า การสแกน (Scanning) ในมอนิเตอร์หากยกตัวอย่างจอแบบ VGA กราฟิกที่เกิดขึ้นกับจอแบบนี้ต้องมีการสแกนให้เกิดเส้นภาพ 480 เส้นภาพต่อ 1 เฟรม

โย้คจะถูกต่อพ่วงเอาไว้ที่ตัวเอาท์พุททรานซิสเตอร์ โดยต่อเข้ากับขาคอลเล็คเตอร์ของทรานซิสเตอร์ภาคฮอริซอนตอลเพื่อให้ทรานซิสเตอร์ส่งพัลส์ออกไปยังโย้ค ให้ห้โย้คเกิดกระแสของสนามแม่เหล็กส่งไปบังคับลำอิเล็กตรอนที่เรียกว่า ลำบีม(Beam)

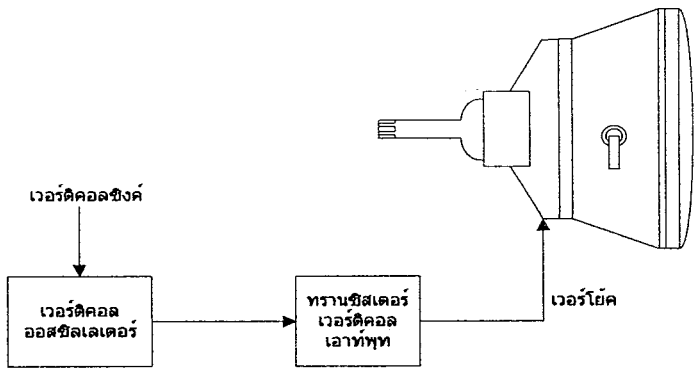
การทำงานของภาคฮอริซอนตอลยังต้องอาศัยข้อมูลหรือสัญญาณที่ส่งมาจากเครื่องคอมพิวเตอร์เพื่อให้เฟรมภาพที่เกิดขึ้นต่อภาพกันได้อย่างถูกต้อง การต่อเรียงภาพได้ถูกต้องเรียกว่าระบบซิงโครไนซ์ เพื่อกำหนดให้ภาคผลิตความถี่ฮอริซอนตอล(Horizontal Oscillator) ทำการผลิตความถี่เริ่มต้น-สิ้นสุดในแต่ละคาบสอดคล้องกับการทำงานของเครื่องคอมพิวเตอร์ แล้วจึงส่งความถี่จากภาคผลิตความถี่มายังภาคขับกระแสหรือฮอริซอนตอลไดรเวอร์ (Horizontal Driver) โดยการทำงานของภาคนี้มีเพาเวอร์ทรานซิสเตอร์ขนาดกลางเป็นอุปกรณ์สำคัญ

ระบบควบคุมการสแกนแนวตั้ง หรือภาคเวอร์ติคอล

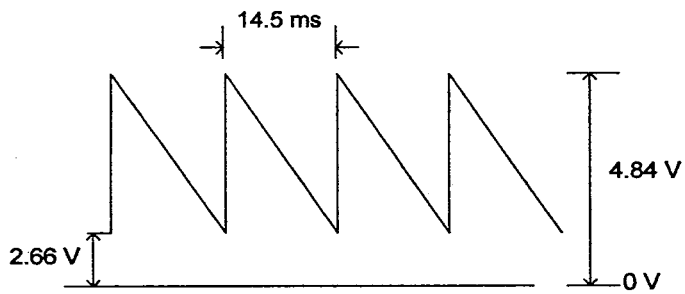
นอกจากการแสกนเพื่อดึงลำอิเล็กตรอนกวาดแสงในแนวนอนเพื่อกำเนิดเส้นภาพแล้ว เส้นภาพที่เกิดขึ้นยังแสกนทับเส้นที่ทับอยู่ การเกิดเฟรมภาพยังเกิดไม่ได้ หากในระบบดีเฟลกชั่นฮอริซอลตอลทำงานได้เพียงอย่างเดียวย่อมเห็นเพียงเส้นเดียวกลางจอทางแนวนอน การบังคับการกวาดลำอิเล็กตรอนจึงต้องกวาดเพื่อไล่เรียงเส้นภาพจากขอบจอด้านบนลงมายังขอบจอทางด้านล่างเพื่อให้ภาพเกิดขึ้นเต็มจอ โดยการลากเส้นแสงต้องเรียงลำดับเส้นอย่างมีระเบียบและต้องทำให้เฟรมภาพในด้านแนวตั้งเหมือนกับเฟรมภาพที่สร้างมาโดยเครื่องคอมพิวเตอร์

เครื่องคอมพิวเตอร์จะส่งสัญญาณเวอร์ติคอลซิงค์มายังภาคผลิตความถี่ของระบบคอมพิวเตอร์อาจมีการเปลี่ยนแปลงมาก-น้อยแล้วแต่ระบบ หากความถี่จากคอมพิวเตอร์ส่งมา 60 เฮิรตซ์ย่อมทำให้ภาคผลิตความถี่เวอร์ติคอล(เวอร์ติคอลออสซิลเลเตอร์) ผลิตความถี่ 60 ครั้งในทุกๆ วินาที

ภาคผลิตความถี่เวอร์ติคอล(ตามรูป)จะผลิตสัญญาณเป็นรูปฟันเลื่อย แล้วจึงส่งไปยังภาคเอาท์พุททำการขับกระแสออกไปยังดีเฟล็กชั่นโย้ค



รูปที่2.4 แผนผังเบื้องต้นของภาคเวอร์ติคอล



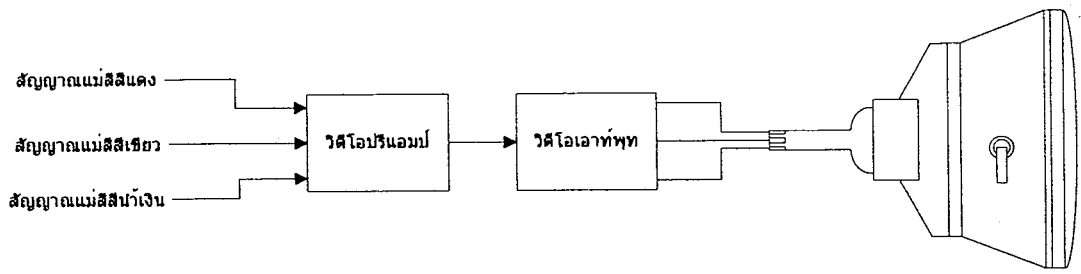
รูปที่2.5 ความถี่ที่ผลิตโดยวงจรเวอร์ติคอล์ออสซิลเลเตอร์

ภาคขยายสัญญาณภาพหรือภาพวิดีโอ

สัญญาณภาพที่ส่งออกไปที่จอของมอนิเตอร์เป็นสัญญาณสีที่เป็นแม่สี 3สี คือ สีแดง เขียว และน้ำเงิน สัญญาณนี้อาจเรียกว่า สัญญาณ RGB ซึ่งหมายถึงตัวย่อของแม่สีทั้ง 3 สี สัญญาณทั้ง 3 สี ส่งมาจากเครื่องคอมพิวเตอร์เพื่อเข้ามาสู่วงจรคอมไบน์และวงจรขยายก่อนส่งออกจอภาพ หากมองภาพของส่วนนี้ให้ง่ายๆดังแผนผังเป็นไปตามรูป 2.6

สัญญาณภาพหรือสัญญาณวิดีโอ(หรือ R G B) จากเครื่องคอมพิวเตอร์ถูกส่งมาด้วยระดับความแรง 0 ถึง 0.7 โวลต์ หน้าที่ของมอนิเตอร์คือการขยายสัญญาณให้มีความแรงมากขึ้น ให้เพียงพอต่อการแสดงผลออกหน้าจอ ในระบบขยายสัญญาณยังมีลักษณะพิเศษบางประการที่เกี่ยวข้องกับระดับแสง, ระดับสี, หรืออื่นๆและมีวงจรหลายภาค

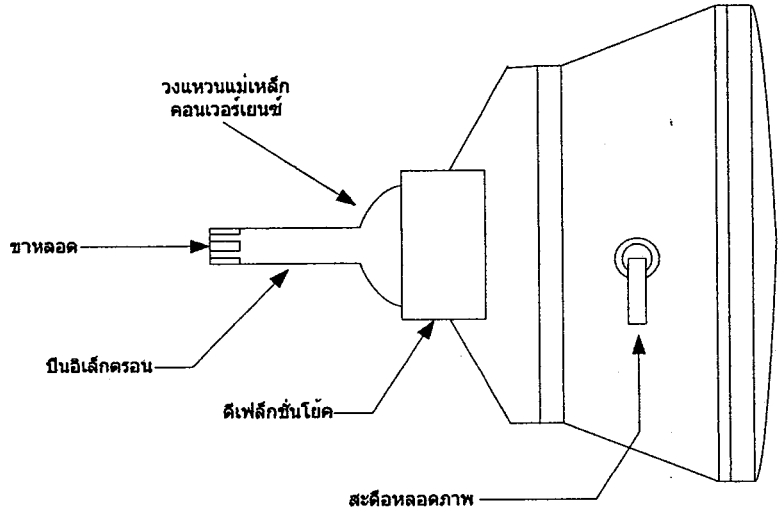
สุดท้ายจึงส่งผลการขยายสัญญาณนั้นออกไปยังทรานซิสเตอร์เพาเวอร์ที่ขับสีแดง สีเขียว และสีน้ำเงิน ซึ่งแยกขับแต่ละวงจรอย่างอิสระเพื่อส่งไปยังคาโถดหลอดภาพของแต่ละสี



รูปที่ 2.6 แผนผังเบื้องต้นของวงจรขยายสัญญาณภาพ

วงจรหลอดภาพ

หลอดภาพเป็นหลอดที่พัฒนามาจากหลอด CRT แบบเดิม เป็นหลอดที่มีหลักการเดียวกับจอของเครื่องรับโทรทัศน์สีโดยมีองค์ประกอบที่สำคัญคือคาโถดเพื่อการยิงลำอิเล็กตรอนของแม่สีทั้ง3สีออกไปยังหน้าจอ การบังคับลำอิเล็กตรอนให้ออกไปมากน้อยขึ้นอยู่กับความต่างศักย์ระหว่างคาโถดกับคอนโทรลกริดหรือกริดอันดับที่ 1(G1) มีสกรีนกริดหรือG2 ทำหน้าที่บังคับลำอิเล็กตรอนให้พุ่ง-เร่งให้วิ่งไปยังจออย่างมีประสิทธิภาพ โดยมีโฟกัสกริดหรือ G3 ควบคุมความคมชัดของลำแสง



รูปที่ 2.7 แผนผังเบื้องต้นของหลอดภาพ

หลักการของหลอดภาพที่ทำให้สามารถจ่ายลำแสงออกไปได้อาศัยความร้อนจากไส้หลอด เมื่อพิจารณาโครงสร้างแล้วจะพบว่าไส้หลอดหรือฟิลาเมนต์ ต้องมีไฟจ่ายเพื่อให้ไส้หลอดร้อน ไส้หลอดไปเผาคาโถดทำให้เกิดสนามไฟฟ้าหรืออิเล็กตรอนฟุ้งขึ้น เมื่อหน้าจอซึ่งมีอาโหนดเป็นอุปกรณ์ไฟสูงจึงทำให้อิเล็กตรอนวิ่งเข้าไปหาอาโหนด ไฟสูงทำหน้าที่เป็นตัวล่อลำอิเล็กตรอน แต่

เมื่อมีไฟเร่งลำอิเล็กตรอนอย่างสกรีนกริด จึงทำให้ลำอิเล็กตรอนวิ่งไปยังหน้าจอได้ ที่หน้าจอเคลือบสารเรียงแสงสีต่างๆเอาไว้ เมื่ออิเล็กตรอนวิ่งชนจึงเกิดการเปล่งแสงขึ้น

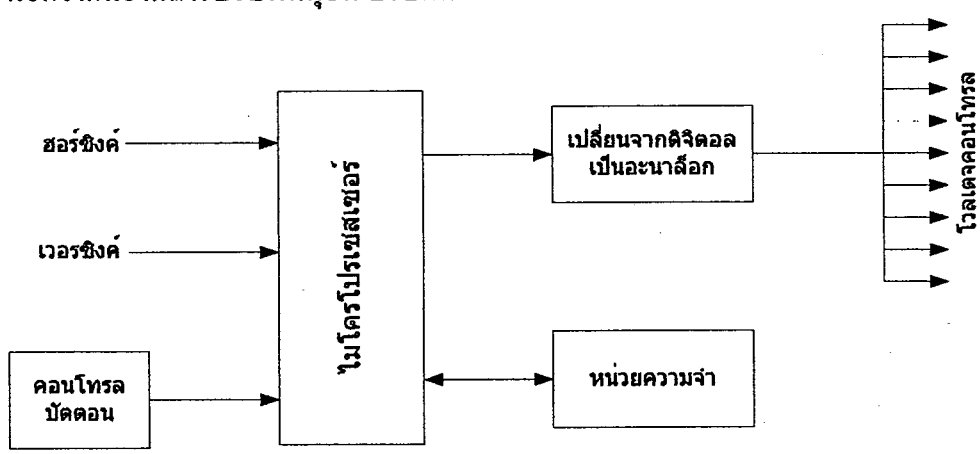
อย่างไรก็ตาม หากไม่มีขดลวดหักเหหรือเบี่ยงเบนลำอิเล็กตรอนที่เรียกว่าโย้ค หรือดีเฟล็กชั่นโย้คมาควบคุมแล้ว แสงที่เกิดขึ้นจะเป็นเพียงจุดเดียวตรงกลางจอ โย้คเข้ามาทำให้เกิดภาพขึ้นมาเต็มทั้งจอ

วงจรควบคุม

มอนิเตอร์ของคอมพิวเตอร์มีระบบควบคุมภาพเพื่อให้เกิดอิมเมจของภาพหลากหลายรูปแบบมากกว่าเครื่องรับโทรทัศน์ ในเครื่องรับโทรทัศน์ทั่วไปจะสามารถปรับแต่งภาพ, แสง, สี, เสียง, หรือความเข้ม ความคมชัด ในส่วนของอิมเมจภาพก็ปรับความสูง, ความเว้า แต่ในมอนิเตอร์มีมากกว่านั้น

มอนิเตอร์ในยุคแรกเป็นการควบคุมปรับแต่งในแบบอะนาล็อกต่อมาเป็นแบบดิจิตอล หรือเป็นการผสมกันของทั้ง 2 อย่าง ในการปรับทางเวอร์ติคอลมีการปรับขนาดจอ, การปรับเซ็นเตอร์หรือตำแหน่งทางแนวตั้ง และการปรับลิเนียร์หรือการตั้งสัดส่วนในมุมต่างๆ ในส่วนของฮอริซอลตอลมีการปรับแต่งซิงค์, ปรับแต่งขนาด, ปรับตำแหน่ง, ปรับเซ็นเตอร์ และปรับสัดส่วน(ลิเนียริตี้)ของมุมต่างๆ การปรับแต่งควบคุมอื่นๆ ก็มีพวกคอนทราสต์,ไบรท์เนสในภาควิดีโอ ในขณะที่ภาคฟลายแบ็คมีการปรับโพกัสและสกรีน

นอกจากนี้ยังมีการปรับพินคุชั่น ปรับแทร็ปพิซอยด์ หรับพาราเลลโลแกรม เป็นต้น



รูปที่ 2.8 แผนผังการควบคุมและไมโครโปรเซสเซอร์ของมอนิเตอร์

หากเป็นการปรับแบบอะนาล็อกจะมีตัวปรับเป็นโพเทนซิโอมิเตอร์หรือVRเป็นตัวปรับ แต่หากเป็นยุคดิจิตอลจะใช้คำสั่งไปยังหน่วยควบคุมภายใน เพื่อส่งสัญญาณไปยังตัว

ไมโครโปรเซสเซอร์และตัวโปรเซสเซอร์จะส่งคำสั่งออกไปยังหน่วยเปลี่ยนดิจิตอลเป็นอะนาล็อก (หรือวงจรD/A) วงจรเปลี่ยนข้อมูลดิจิตอลเป็นอะนาล็อก จะแปลงข้อมูลให้เป็นแรงดันไฟไปควบคุมมอนิเตอร์ไปตามรูป 2.8

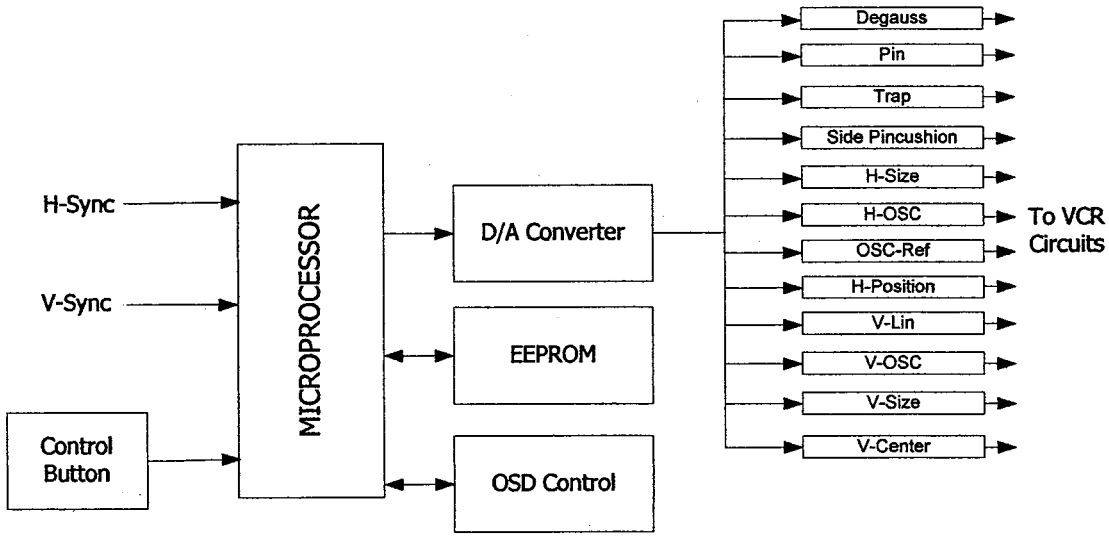
ภาคไมโครโปรเซสเซอร์

รูปแบบของไมโครโปรเซสเซอร์ 2.9 จะประกอบด้วยส่วนต่างๆที่สำคัญคือ ไมโครโปรเซสเซอร์ วงจร D/A คอนเวอร์เตอร์ และชิพที่เป็นหน่วยความจำ

โดยหลักๆแล้วไมโครโปรเซสเซอร์เป็นภาคที่จะเข้าไปจัดการกับสัญญาณควบคุมในแบบดิจิตอล และต้องรับสัญญาณซิงค์ทางเวอร์ติคอลและฮอริซอลตอลเพื่อมาผนวกเข้ากับระบบของมอนิเตอร์ให้ทำงานอัตโนมัติ สอดคล้องกับความต้องการของข้อมูลที่จะแสดงผล นอกจากนั้นแล้วต้องรองรับระบบควบคุมแบบอัตโนมัติ ในการปรับแต่ง ควบคุมกระแสของวงจรด้านภาพ

หลังจากกระบวนการทางอินพุตผ่านไป ไมโครโปรเซสเซอร์จะส่งสัญญาณออกทางเอาท์พุต แต่เนื่องจากข้อมูลหรือสัญญาณเป็นข้อมูลดิจิตอล จึงต้องผ่านวงจรแปลงดิจิตอลเป็นอะนาล็อก ที่เรียกว่าวงจร D/A คอนเวอร์เตอร์ เพื่อจ่ายเข้าสู่วงจรควบคุมต่อไป

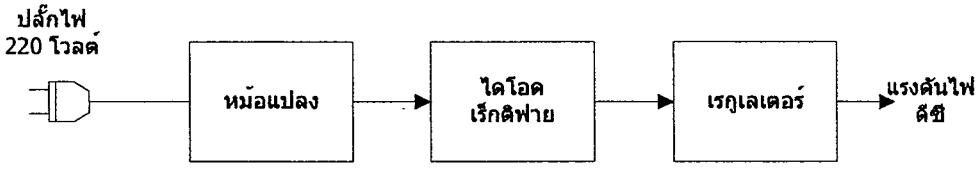
ส่วนหน่วยความจำเป็นชิพไอซีที่จะเก็บข้อมูลต่างๆที่เราตั้ง-ปรับข้อมูลให้กับจอมอนิเตอร์ ชิพหน่วยความจำเสมือนทำหน้าที่เป็นหน่วย "รีเมมเบอร์" เมื่อตอนที่เราปิดเครื่อง และเมื่อเราเปิดเครื่องขึ้นมาใหม่ เราเคยตั้ง-ปรับแต่งข้อมูลใดๆไว้ มันจะคงข้อมูลการปรับแต่งอันนั้น สนองตอบเราเหมือนตอนก่อนการเปิดเครื่องครั้งสุดท้าย



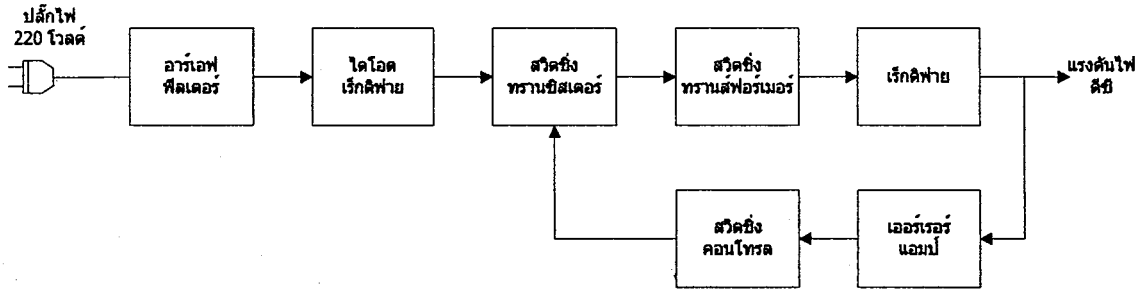
รูปที่ 2.9 แผนผังของไมโครโปรเซสเซอร์และวงจรสนับสนุน

ภาคจ่ายไฟ

ภาคจ่ายไฟโดยรวมแล้วก็คือภาคจ่ายแบบสวิตโหมดเพาเวอร์ซัพพลาย แต่อาจจะมียุคแรกๆ ที่เป็นลิเนียร์เรกูเลเตอร์ แต่ลิเนียร์เรกูเลเตอร์มีการสูญเสียด้านความร้อนสูงทำให้เปลืองพลังงาน เมื่อเครื่องคอมพิวเตอร์ถูกใช้งานมากในงานสำนักงาน ทำให้เลือกใช้เป็นภาคจ่ายไฟแบบสวิตชิ่งซึ่งประหยัดและทนทาน



รูปที่ 2.10 แผนผังของลิเนียร์เรกกูเลเตอร์



รูปที่ 2.11 แผนผังของสวิตชิ่งโหมดเพาเวอร์ซัพพลาย

บล็อกไดอะแกรมหรือแผนผังของภาคจ่ายไฟแบบลิเนียร์เรกูเลเตอร์เป็นไปดังรูป 2.10 และภาคจ่ายไฟแบบสวิตโหมดเป็นไปดังรูป 2.11

สำหรับภาคจ่ายไฟในจอมอนิเตอร์ใกล้เคียงกับอุปกรณ์ชนิดอื่น แต่จะมีความสามารถเพิ่มเติม เช่น จ่ายไฟได้หลากหลายเพื่อเซฟการทำงานของจอมอนิเตอร์ หากเครื่องไม่ได้ใช้งานจะตัดระบบจ่ายไฟหลักออกไปเป็นหลักการของเพาเวอร์เซฟ หรือกรีนเซฟอีกทางหนึ่ง

2.2FPGA

ในปัจจุบันมี FPGA อยู่ 4 ชนิดที่วางขายอยู่ในท้องตลาดได้แก่ Symmetrical Array, Row-Based, Hierarchical PLD และ Sea-of-Gates ซึ่งแต่ละชนิดก็มีลักษณะการเชื่อมต่อภายในและการโปรแกรม ที่แตกต่างกันไป นอกจากนี้ในการแบ่งประเภทของ FPGA อาจแบ่งได้ตามเทคโนโลยีที่ใช้ในการโปรแกรม ซึ่งมีอยู่ 2 แบบคือ การโปรแกรมโดยการทำให้เกิดการเปลี่ยนแปลงทางกายภาพของตัวชิพ และการโปรแกรม โดยการใช้หน่วยความจำ

2.2.1 การโปรแกรมโดยการทำให้เกิดการเปลี่ยนแปลงทางกายภาพ

2.2.1.1 Fuse เป็นวิธีการโปรแกรมที่สามารถทำได้เพียงครั้งเดียว ซึ่งหลังจากที่โปรแกรมแล้วจุดเชื่อมต่อจะขาดจากกัน

2.2.1.2 Anti Fuse เป็นวิธีการโปรแกรมที่คล้ายกับแบบ Fuse แต่ต่างกันที่หลังจากทำการโปรแกรม แล้วจุดเชื่อมต่อจะเชื่อมถึงกัน

2.2.2 การโปรแกรมโดยใช้หน่วยความจำ

2.2.2.1 EEPROM Based FPGA

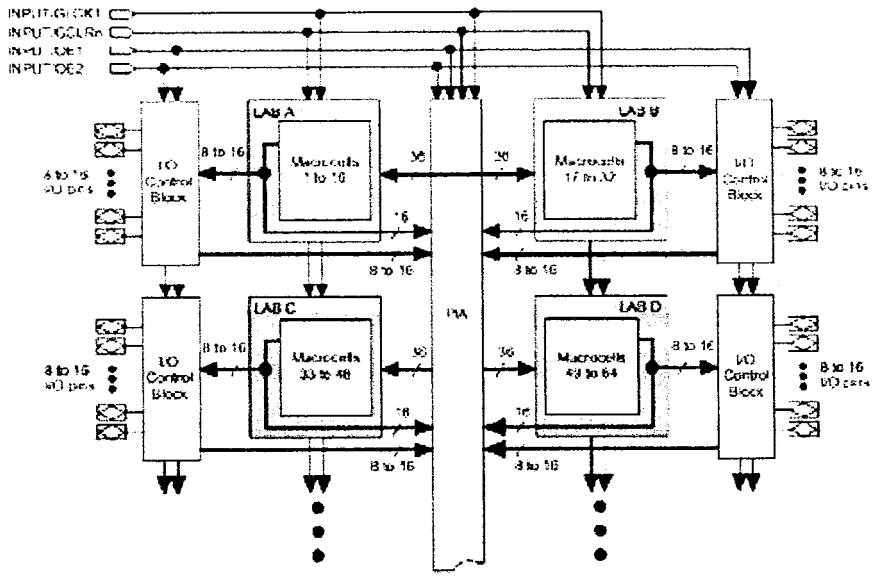
FPGA ที่ใช้การโปรแกรมแบบนี้มักเรียกว่า CPLD ซึ่งเทคโนโลยีที่ใช้จะเหมือนกับ EEPROM ทำให้มีความจุของเกทต่ำ โดยทั่วไปจะน้อยกว่า 20,000 เกท แต่ข้อดีของ EEPROM Based FPGA คือสามารถเก็บข้อมูลที่โปรแกรมลงไปได้โดยไม่จำเป็นต้องมีไฟเลี้ยง และในการโปรแกรมจะใช้ทรานซิสเตอร์ 1 ตัวต่อ 1 บิท ซึ่งการโปรแกรมสามารถทำได้ประมาณ 10,000 ครั้ง

2.2.3 SRAM Based FPGA

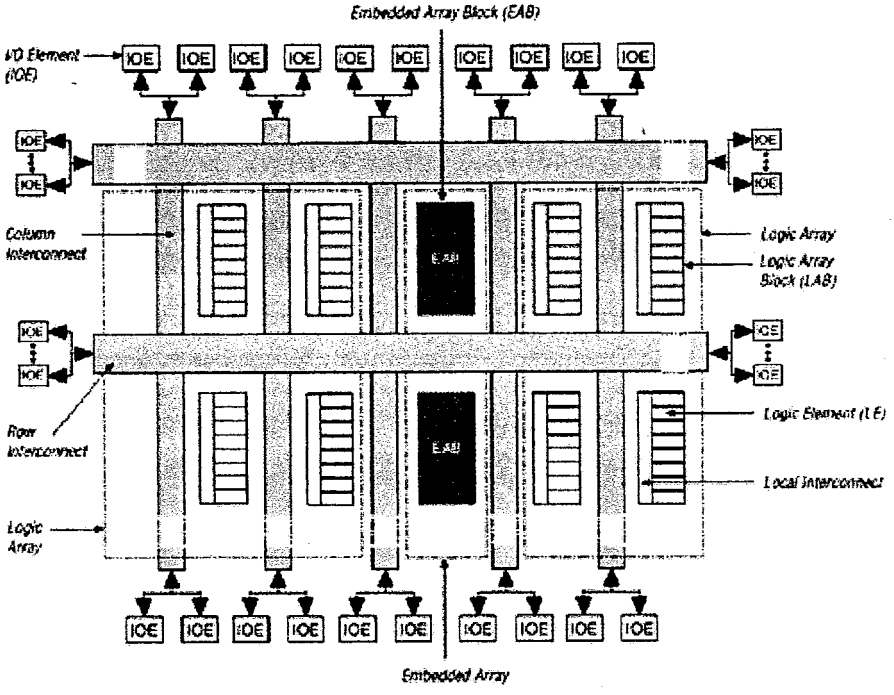
FPGA แบบนี้จะใช้เทคโนโลยีในการโปรแกรมเหมือนกับ SRAM (Static RAM) ทำให้สามารถ โปรแกรมซ้ำได้โดยไม่จำกัดจำนวนครั้ง นอกจากนี้ยังมีความจุของเกทในระดับปานกลางถึงสูงมาก (ประมาณ 10,000 - 1,000,000 เกท) ซึ่งข้อดีของ SRAM BasedFPGA คือใช้เวลาในการโปรแกรมน้อย (ระดับ nsec) การโปรแกรมทำได้ง่ายเทียบได้กับการเขียน SRAM ทั่วไป และเหมาะสำหรับการออกแบบวงจรที่มีความสลับซับซ้อน ส่วนข้อเสียคือไม่สามารถเก็บโปรแกรมในภาวะที่ไม่มีไฟเลี้ยงได้ ดังนั้น FPGA ชนิดนี้จึงมักใช้ควบคู่กับ ROM เพื่อเก็บโปรแกรมและทำการโหลดโปรแกรมลงในตัวชิพในขณะที่เริ่มต้นใช้งาน

โครงสร้างภายในของ FPGA

ลักษณะโครงสร้างภายในของ FPGA จะเป็นอาเรย์ของบล็อกลอจิกที่สามารถทำการโปรแกรมได้ดังรูปที่ 2.12 และ 2.13



รูปที่ 2.12 โครงสร้างภายในของ FPGA ตระกูล MAX7000S



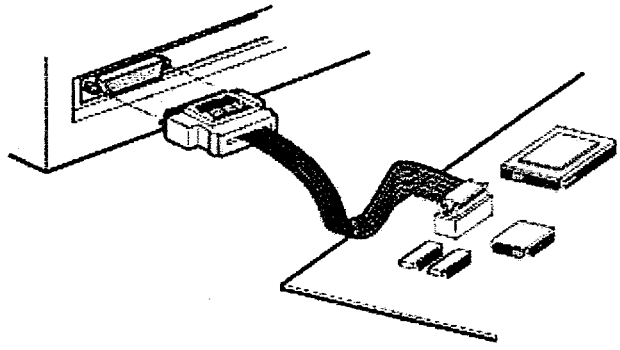
รูปที่ 2.13 โครงสร้างภายในของ FPGA ตระกูล FLEX10K

ข้อดีของการออกแบบด้วย FPGA

1. ผู้ออกแบบไม่จำเป็นต้องทราบถึงโครงสร้างภายในของตัวชิพ เพียงแต่มีความรู้เกี่ยวกับขั้นตอนการออก แบบลอจิกก็เพียงพอแล้ว ต่างกับการใช้ไมโครโปรเซสเซอร์ซึ่งจำเป็นต้องศึกษาโครงสร้างภายใน รวมถึงภาษาแอสเซมบลี้(Assembly) ของไมโครโปรเซสเซอร์ตัวนั้นด้วย

2. มีการออกแบบโดยใช้ภาษาในการอธิบายการทำงานของวงจร หรือ HDL (Hardware Description Language) เป็นเครื่องมือในการออกแบบ ซึ่งเป็นวิธีการที่มีความยืดหยุ่นสูง ทำได้รวดเร็ว และไม่จำเป็น ต้องทราบถึงลักษณะของวงจรที่ต้องการว่าจะเชื่อมต่อกันอย่างไร เพียงแต่กำหนดลักษณะการทำงานให้มัน จากนั้นตัวซอฟต์แวร์จะทำ Synthesis and Optimize ให้ทั้งหมด นอกจากนี้ภาษาที่ใช้ยังเป็นมาตรฐานเดียวกัน สามารถใช้ได้กับชิพทุกตัวและทุกบริษัท

3. การโปรแกรมสามารถทำได้เองและใช้เวลาไม่นาน เพียงแค่ส่งข้อมูลผ่านสายดาวน์โหลดทางพอร์ตของ คอมพิวเตอร์ก็สามารถโปรแกรมตัวชิพขณะที่อยู่ในระบบได้ โดยไม่จำเป็นต้องถอดมาโปรแกรมข้างนอก ดังรูปที่ 2.3 และที่สำคัญสามารถโปรแกรมได้หลายครั้ง จึงทำให้ง่ายในการแก้ไขและพัฒนาโดยไม่เสียค่าใช้จ่ายเพิ่มเติมแต่อย่างใด



รูปที่ 2.14 การโปรแกรมลงในชิพ

การออกแบบโดยใช้ภาษาอธิบายพฤติกรรมของฮาร์ดแวร์

ในการออกแบบวงจรดิจิตอลนั้นสามารถทำได้โดยการวาดวงจร (Schematic) หรือใช้ภาษาอธิบายพฤติกรรม (Hardware Description Language) ของฮาร์ดแวร์ จากที่ได้กล่าวไปแล้วในบทที่ 1 ในกรณีของการออก แบบวงจรด้วย ASIC ชนิด Full Custom ผู้ออกแบบจะต้องเขียนวงจรด้วย Schematic จากนั้นจะนำวงจรที่ ออกแบบไว้ไปทำการจำลองการทำงาน (Simulate) ซึ่งหากผลออกมาเป็นที่พอใจก็จะต้อง Layout เป็นชั้นสาร และในการออกแบบ ASIC ชนิดนี้ผู้ออกแบบจำเป็นจะต้องทราบถึงเทคโนโลยีที่ใช้ในการสร้างด้วย หลังจากได้ Layout ที่สมบูรณ์แล้วจึงจะ

ส่งไปเข้ากระบวนการสร้างไอซีหรือ Fabrication เพื่อสร้างเป็นชิพไอซีออกมา แต่ในการออกแบบวงจรด้วย FPGA โดยการใช้ Schematic หรือใช้ภาษาอธิบายการทำงานของวงจรจะทำ ได้สะดวกกว่า เนื่องจากวิธีการนี้ผู้ออกแบบไม่จำเป็นต้องคำนึงถึงเทคโนโลยีที่จะใช้สร้างไอซีและที่สำคัญการออก แบบโดยวิธีนี้สามารถแก้ไขโมเดล (Model) หรือเปลี่ยนแปลงเทคโนโลยีได้สะดวกกว่า เพราะไม่ต้องวาดวงจร ใหม่ นั่นคือการออกแบบโดยใช้ภาษาอธิบายฮาร์ดแวร์ จะทำให้โมเดลที่ได้ไม่ขึ้นกับเทคโนโลยีสำหรับภาษาที่ใช้ สำหรับอธิบายพฤติกรรมของฮาร์ดแวร์ที่ใช้กันก็มี VHDL,ADHL และ Verilog เป็นต้น ส่วนรายละเอียดของขั้นตอนในการออกแบบสามารถอธิบายได้ดังนี้

1.การสังเคราะห์วงจร(Logic Synthesis)

ในขั้นตอนนี้จะใช้ซอฟต์แวร์ในการสังเคราะห์วงจร (Synthesis Tools) ทำการสังเคราะห์พฤติกรรม ของวงจรที่ได้จากการออกแบบด้วย Schematic หรือ VHDL ซึ่งต้องทำการตรวจสอบด้วยว่าซอฟต์แวร์ นั้นสนับสนุนเทคโนโลยี FPGA (FPGA Library) ที่ต้องการหรือไม่ ตัวอย่างเช่น FPGA ของบริษัท XILINX และบริษัท ALTERA จะมีซอฟต์แวร์หลายตัวที่สามารถใช้ได้ เช่น Max Plus II ในขั้นตอนนี้ ซอฟต์แวร์สังเคราะห์วงจรจะทำการแปลงโค้ด VHDL และทำการ Optimize เพื่อให้ได้วงจรตาม เทคโนโลยีที่เลือกใช้ในการสังเคราะห์วงจรนั้นวงจรระดับเกท (Gate Level) จะไม่เหมาะสมกับโครง สร้างที่มีอยู่ในอุปกรณ์ FPGA ดังนั้นในการ Optimize ซอฟต์แวร์สังเคราะห์วงจร จะต้องทำการ Optimize ให้ได้เป็นวงจรที่ประกอบ ด้วยกลุ่มของลอจิกที่เหมาะสมกับอุปกรณ์ FPGA นั้นๆจึงทำให้ผล ที่ได้มีประสิทธิภาพและในขั้นตอนการสังเคราะห์วงจรนี้ ผู้ออกแบบสามารถกำหนดข้อบังคับสำหรับโมเดล แต่ละตัวได้ เช่น ข้อบังคับในเรื่องเวลา (Timing Constraints) หรือข้อบังคับในเรื่องของพื้นที่ (Area) หรือกำหนดชนิดและตำแหน่งของ I/O ซึ่งข้อบังคับเหล่านี้จะถูกนำไปใช้ในขั้นตอน Optimize เพื่อให้วงจร ที่ได้เป็นไปตามที่กำหนด ส่วนสำคัญในการ Optimize คือการเทียบ (Mapping) โมเดลให้เข้ากับ เทคโนโลยีที่ใช้เพื่อให้ได้วงจรที่เหมาะสมกับโครงสร้างและสถาปัตยกรรมภายในอุปกรณ์ FPGA เมื่อทำ การสังเคราะห์วงจรเสร็จแล้ว ซอฟแวร์การสังเคราะห์วงจรก็จะมีการรายงานผลว่าโมเดลที่ออกแบบไปนั้น เป็นอย่างไร เช่น มีค่าความหน่วง (Delay) เท่าใด ใช้ทรัพยากรต่างๆใน FPGA อะไรบ้าง เมื่อมาถึงขั้น ตอนนี้ผู้ออกแบบก็จะทราบว่าโมเดลเป็นไปตามข้อบังคับหรือไม่ ถ้าไม่ก็สังเคราะห์ใหม่จนกว่าจะเป็นไปตามที่กำหนด

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.การแบ่งวงจร(Partitioning)

ขั้นตอนนี้เป็นการแบ่งวงจรที่ได้จากการสังเคราะห์ เป็นส่วนย่อยๆ สำหรับลงใน CLBs, IOBs หรือองค์ ประกอบอื่นๆ ภายในอุปกรณ์ FPGA สำหรับเกณฑ์ที่ใช้ในการแบ่งคือให้แต่ละส่วนที่จะแยกออกจากกัน มีจำนวนสัญญาณที่เชื่อมต่อระหว่างกันน้อยที่สุดเท่าที่จะทำได้ เพื่อลดความหนาแน่นในตอนทำการเชื่อม ต่อสัญญาณ (routing) ในขั้นตอนนี้จะใช้ซอฟต์แวร์ทำโดยซอฟต์แวร์จะเทียบส่วนประกอบของวงจรเช่น เกท (gate), ฟลิป-ฟลอบ (flip-flop) ลงในทรัพยากรต่างๆ ที่มีอยู่ในอุปกรณ์ FPGAหลังจากทำขั้นตอน นี้เสร็จแล้วผู้ออกแบบสามารถที่จะทราบว่าวงจรใช้จำนวนทรัพยากรภายในอุปกรณ์ FPGA ไปเท่าไร ส่วนข้อมูลทางเวลานั้นผู้ออกแบบจะทราบเฉพาะความหน่วงภายในแต่ละส่วนเท่านั้น หรือที่เรียกว่าความ หน่วงลอจิก(logic delay) ส่วนซอฟต์แวร์จะรวมเอาซอฟต์แวร์ย่อยอื่นๆอีก เพื่อให้การทำ PPR(Partitioning Placement & Routing)เป็นไปต่อเนื่อง

3.การอุปกรณ์(Placement)

ขั้นตอนนี้เป็นการเลือกทำเลที่ตั้งของแต่ละส่วนของวงจรที่ผ่านการแบ่งวงจร (Partitioning) มาแล้วว่าควร จะอยู่ ณ ตำแหน่งไหนในอุปกรณ์ FPGA เพื่อให้ได้ผลลัพธ์ดีที่สุด เช่นวงจรส่วนไหนควรอยู่ใกล้กัน เพื่อจะ ได้ค้นหาเส้นทางได้ (route) ง่ายหรือช่วยลดความหน่วง จะเห็นได้ว่าตำแหน่งภายในอุปกรณ์ FPGA นั้นมี ความสำคัญเพราะถ้าจัดวางวงจรลงในตำแหน่งที่ไม่เหมาะสมแล้วจะทำให้ความหน่วงเพิ่มขึ้นหรือ Router ทำการค้นหาเส้นทางสัญญาณไดไม่หมด การวางอุปกรณ์ที่ดีควรวางส่วนต่างๆให้อยู่ใกล้กันโดยเฉพาะส่วน ที่มีการเชื่อมต่อสัญญาณด้วยกันนอกจากนั้นการกำหนดตำแหน่งขา I/O (I/O pin) ตามตำแหน่งขา I/O ของ FPGA บนแผ่น PCB ก็จะมีผลโดยตรงเลยคือซอฟต์แวร์จะวาง I/O ลงในตำแหน่งที่ผู้ออกแบบกำหนด ซึ่ง บางครั้งตำแหน่งที่กำหนดไปไม่เหมาะสม ดังนั้นการกำหนดขา I/O ควรกำหนดตำแหน่งให้เหมาะสม หรือ ไม่ก็ให้ซอฟต์แวร์จัดการเอง

4.การเชื่อมต่อสัญญาณ(Routing)

ในขั้นตอนนี้เป็นการเชื่อมต่อสัญญาณระหว่างองค์ประกอบต่างๆ ภายในอุปกรณ์ FPGA ขั้นตอนนี้จะทำ ต่อเนื่องจากการวางอุปกรณ์ ในกรณีที่ทำการวางอุปกรณ์ไว้ไม่ดีซอฟต์แวร์ก็จะทำการเชื่อมต่อสัญญาณ ได้ไม่หมด (เนื่องจากจำนวนทรัพยากรสำหรับเชื่อมต่อสัญญาณนั้นมีอยู่จำกัด) หรือเกิดความหน่วงเกิน ค่าที่กำหนดในข้อบังคับ ผู้ออกแบบสามารถทำขั้นตอนนี้ได้โดยใช้ซอฟท์แวร์หรือผู้ออกแบบจะทำการ เชื่อมต่อสัญญาณด้วยตนเองก็ได้ แต่ทางที่ดีควรใช้ซอฟต์แวร์ทำดีกว่า นอกจากนั้นการกำหนดข้อบังคับทางเวลา จะช่วยให้ผลที่ได้จากการเชื่อมต่อสัญญาณดีขึ้นได้

61334

5.ความหน่วงด้านเวลา(Delay)

ในการทำ FPGA นั้นความหน่วงที่เกิดขึ้นเป็นความหน่วงที่เกิดจากการวางตำแหน่ง (layout) ของอุปกรณ์ ซึ่งผู้ออกแบบไม่สามารถเข้าไปแก้ไขได้ แต่สามารถทำให้มีความหน่วงน้อยที่สุดได้ สำหรับความหน่วงที่เกิดขึ้นนั้นแยกเป็นสองประเภท คือ

- ความหน่วงลอจิก(Logic Delay)เป็นความหน่วงภายในองค์ประกอบของอุปกรณ์ FPGA เอง

-ความหน่วงที่เกิดจากการเชื่อมต่อสัญญาณ(Routing Delay)เป็นความหน่วงที่เกิดจากการเชื่อมต่อสัญญาณระหว่างองค์ประกอบภายในอุปกรณ์ FPGA

โดยปกติแล้ว ค่าความหน่วงลอจิกไม่ควรเกิน 50% ของค่าความหน่วงที่ยอมรับได้ เพราะความหน่วงที่เกิดจากการ เชื่อมต่อสัญญาณมักจะมีค่ามากกว่าค่าความหน่วงลอจิก ดังนั้นในการวางอุปกรณ์ และเชื่อมต่อสัญญาณ ผู้ออกแบบควรกำหนดข้อบังคับทางเวลาเพื่อให้ ซอฟต์แวร์ได้ทำงานอย่างมีประสิทธิภาพเพิ่มขึ้น และเพื่อ ให้ได้ผลลัพธ์ที่ดีขึ้นค่าความหน่วงที่ได้หลังจากการวางอุปกรณ์ และเชื่อมต่อสัญญาณแล้วจะมีค่าความ หน่วงที่ค่อนข้างแน่นอน ซึ่งผู้ออกแบบสามารถทราบได้ว่าโมเดลที่ออกแบบนั้น เป็นไปตามข้อกำหนดหรือไม่

6.การจำลองการทำงานของวงจร(Simulation)

ในขั้นตอนนี้เป็นขั้นตอนที่สำคัญอีกขั้นตอนหนึ่ง เพราะเป็นขั้นตอนที่ผู้ออกแบบตรวจสอบฟังก์ชันการ ทำงานของโมเดลว่าถูกต้องหรือไม่ มีข้อผิดพลาดตรงไหนเพื่อจะได้ทำการแก้ไขให้ถูกต้อง ในขั้นตอนนี้ จะมีซอฟต์แวร์ที่ใช้สำหรับทำการจำลองการทำงานของวงจรที่ใช้อยู่ เช่น Model Sim ของบริษัท Model Technology หรือ Max Plus II ของบริษัท Altera ในการจำลองการทำงานของวงจร ควรทำทุกครั้งหลังจากที่มีการทำแต่ละขั้นตอนหลักเสร็จแล้ว เพื่อจะได้ทราบว่าข้อผิดพลาดของโมเดล เกิดขึ้นตอนไหน จะได้แก้ไขข้อผิดพลาดตรงขั้นตอนนั้นๆ ได้เลย ไม่ต้องมาคอยตรวจหาขั้นตอนที่ทำให้ เกิดข้อผิดพลาด นั่นคือการทำการจำลองการทำงานของวงจร ต้องทำทั้งหลังการเขียนโค้ด, การสังเคราะห์วงจร และการทำ PPR การจำลองการทำงานของวงจรหลังจากที่เขียนโค้ดเสร็จแล้วนั้น ผู้ออกแบบสามารถทราบได้แค่โมเดลทำงานถูกต้องหรือไม่เท่านั้น (functional test) ยังไม่สามารถตรวจสอบการทำงานในเชิงเวลาได้ถูกต้อง ในการจำลองการทำงานของวงจรหลังจากที่สังเคราะห์เป็นวงจรแล้วเพื่อตรวจสอบว่าฟังก์ชันการทำงานยังคงถูกต้องหรือไม่ และค่าความหน่วงที่เกิดขึ้นเป็นไปตามข้อบังคับหรือไม่ มีข้อผิดพลาดเกิดขึ้นหรือไม่ถ้ามีจะแก้ไขให้ถูกต้อง

ในการจำลองการทำงานของวงจรหลังจากที่ทำการวางอุปกรณ์ การเชื่อมต่อสัญญาณ (Post layout simulation) แล้วก็มีความสำคัญเช่นกันเพราะผลที่ได้จากการจำลองการทำงานของวงจร

ในตอนนี้ จะเป็นผลลัพธ์ของโมเดลเลย ซึ่งผู้ออกแบบนอกจากจะตรวจสอบฟังก์ชันการทำงานแล้ว ยังต้อง ตรวจสอบคุณสมบัติอื่นๆ เช่น ความหน่วงที่ได้จากการทำ PPR ในรูปแบบค่าความหน่วงมาตรฐาน (Standard Delay Format : SDF) ว่าตรงตามที่กำหนดหรือไม่ หรือตรวจสอบว่าวงจรรวมสามารถใช้งานที่ความถี่สูงสุดเท่าไรนั่นเอง ในการจำลองการทำงาน ของวงจรควรใช้ ซอฟต์แวร์ตัวเดียวกันตลอดเพื่อจะได้เปรียบเทียบผลที่ได้จากขั้นตอนต่างๆ

7.การโปรแกรมอุปกรณ์FPGA(Configuration)

หลังจากที่โมเดลผ่านขั้นตอนต่างๆ จนกระทั่งผ่านการทำ PPR (Partitioning, Placement & Routing) แล้วนั้น ถึงตอนนี้ก็สามารถที่จะดาวน์โหลด (download) ลงในอุปกรณ์ FPGA ได้แล้ว ในการดาวน์โหลดนี้ก่อนอื่นต้องแปลงแบบวงจรรวมที่ได้เป็นข้อมูลวงจร (configuration data) ซึ่งอยู่ในรูปของบิตสตรีม (bit stream) ก่อนแล้วจึงดาวน์โหลดลงไปเพื่อให้อุปกรณ์ FPGA มี ฟังก์ชันการทำงานตามโมเดลที่ผู้ออกแบบต้องการ ซึ่งในขั้นตอนนี้จะใช้วิธีที่แตกต่างกันออกไปสำหรับอุปกรณ์ FPGA ของแต่ละบริษัทผู้ผลิตคือ ในกรณีที่เป็นอุปกรณ์ FPGA ชนิดที่ต้องโปรแกรมโดยวิธี SRAM นั้น ในการใช้งานผู้ออกแบบจะต้องเก็บข้อมูลวงจรไว้ในหน่วยความจำประเภท EPROM หรือ serial PROM ด้วยเพื่อจะใช้งานสะดวกขึ้น คือในการใช้งานโมเดลครั้งต่อไปไม่ ต้องดาวน์โหลดข้อมูลวงจรจากเครื่องคอมพิวเตอร์อีก เพราะมีข้อมูลวงจรเก็บอยู่ในหน่วยความจำอยู่แล้ว แต่กรณีที่อุปกรณ์ FPGA เป็นชนิดที่โปรแกรมโดยใช้วิธี EPROM หรือ Anti fuse ก็ไม่จำ เป็นต้องมีหน่วยความจำสำหรับเก็บข้อมูลวงจร เพราะว่าอุปกรณ์ FPGA ชนิดนี้เมื่อดาวน์โหลดข้อมูลวงจรลงไป ข้อมูลที่ดาวน์โหลดลงไปก็ยังคงอยู่ในอุปกรณ์ FPGA และครั้งต่อไปก็ใช้งานโมเดลที่ออกแบบไว้ได้เลย

เครื่องมือสำหรับการออกแบบFPGA

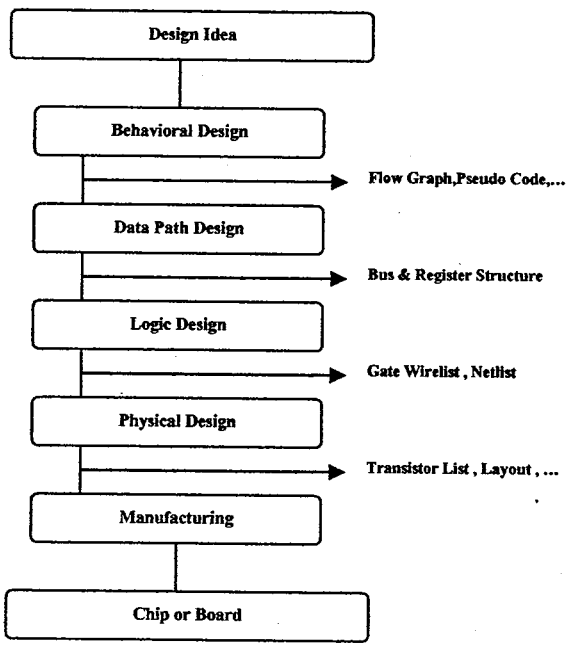
จะเห็นได้ว่าการออกแบบเพื่อทำ FPGA นั้นทำได้สะดวกกว่า ASIC มากเพราะใช้เวลาน้อยกว่ามากด้วย ส่วน สำคัญที่ใช้ในการทำ FPGA คือ ซอฟต์แวร์ที่ใช้ตั้งแต่เขียน โค้ดอธิบายฮาร์ดแวร์จนกระทั่วดาวน์โหลดลงใน อุปกรณ์ FPGA ซึ่งซอฟต์แวร์ที่ใช้ต้องเป็น ซอฟต์แวร์ที่ทำงานต่อเนื่องกันได้ สำหรับซอฟต์แวร์ที่ใช้ทำการ จำลองการทำงานของวงจรนั้น ต้องสามารถใช้งานต่อเนื่องกับซอฟต์แวร์ที่ใช้ทั้งระบบ เพราะโมเดลที่ได้จาก การทำขั้นตอนต่างๆ (ด้วยซอฟต์แวร์ต่างๆ ต้องเอามาจำลองการทำงานได้ และในการจำลองการทำงานของ วงจรควรใช้ซอฟต์แวร์ตัวเดียวกันตลอดทั้งระบบ เพื่อจะได้เปรียบเทียบผลได้ง่าย ในอดีตซอฟต์แวร์ส่วนใหญ่ จะใช้งานอยู่บนคอมพิวเตอร์สมรรถนะสูงอย่างเวิร์กสเตชั่น (Workstation) ในปัจจุบันมีการพัฒนาซอฟต์แวร์ ที่ใช้บนพีซี(PC)มากขึ้น ซึ่งสามารถลดค่าใช้จ่ายในด้านอุปกรณ์คอมพิวเตอร์

2.3การออกแบบทางด้านซอร์ฟแวร์โดยภาษาVHDL

ความซับซ้อนและขนาดของระบบดิจิตอลในปัจจุบันได้เพิ่มมากขึ้นทุกขณะ ส่งผลให้มีการนำคอมพิวเตอร์เพื่อช่วยในการออกแบบหรือ CAD มาใช้ในขบวนการออกแบบฮาร์ดแวร์เพิ่มขึ้นเช่นกัน อีกทั้งอุปกรณ์และวิธีการออกแบบใหม่ๆก็ถูกพัฒนาขึ้นมาเพื่อช่วยอำนวยความสะดวกให้กับนักออกแบบมากขึ้นด้วย สำหรับภาษาบรรยายอุปกรณ์ฮาร์ดแวร์ (HDL: Hardware Description Language) ก็เป็นเครื่องมืออย่างหนึ่งที่ได้รับการพัฒนามาอย่างต่อเนื่อง เพื่อช่วยให้การปรับปรุงขบวนการออกแบบระบบดิจิตอลเป็นไปอย่างมีประสิทธิภาพ

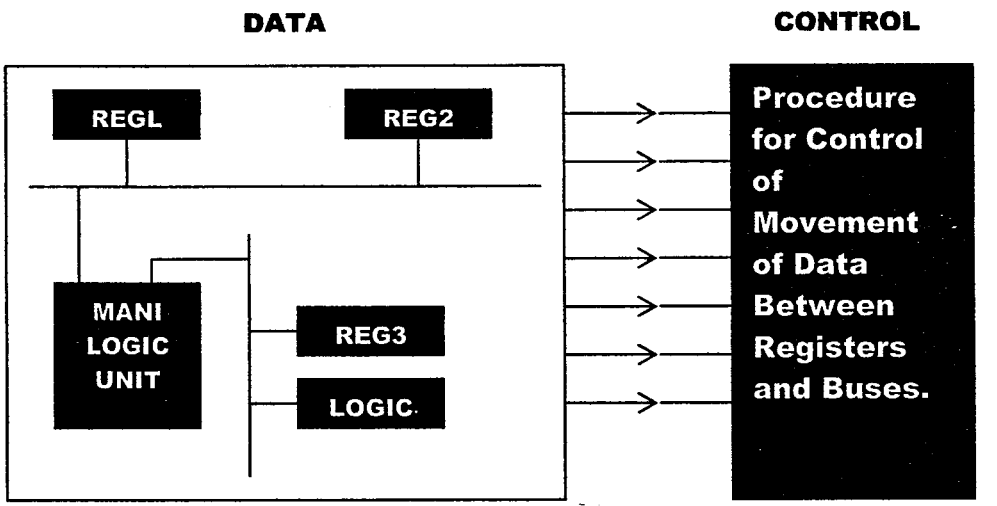
2.3.1 การออกแบบระบบดิจิตอล

ในการออกแบบระบบดิจิตอล เริ่มตั้งแต่การกำหนดแนวความคิดเบื้องต้นจนกระทั่งได้ออกมาเป็นอุปกรณ์ฮาร์ดแวร์ที่ใช้งานได้ต้องผ่านขั้นตอนต่างๆมากมายและในแต่ละขั้นตอนผู้ออกแบบจะต้องตรวจสอบผลลัพธ์ในแต่ละขั้น ก่อนเข้าสู่กระบวนการออกแบบในขั้นต่อไป รูปที่ 2.15 แสดงขั้นตอนปกติที่ใช้ในการออกแบบระบบดิจิตอลทั่วไป ขั้นแรก ผู้ออกแบบจะกำหนดแนวความผิดในการออกแบบแล้วทำการพัฒนาให้สามารถนำมาใช้อย่างสมบูรณ์ ซึ่งภายในขั้นตอนนี้ผู้ออกแบบจำเป็นต้องสร้างรูปแบบระบบในเชิงพฤติกรรมขึ้นมาตรวจสอบซึ่งอาจจะเป็นผังงานแสดงแบบหรือรหัสคำสั่งเทียม(Pseudo Code)ก็ได้



รูปที่ 2.15 ขั้นตอนการออกแบบระบบดิจิตอล

ขั้นตอนต่อไปเป็นการออกแบบระบบเส้นทางของข้อมูล ผู้ออกแบบจะกำหนดส่วนประกอบของรีจิสเตอร์และวงจรลอจิก ที่จำเป็นทั้งหมดเพื่อนำมาประกอบเป็นระบบที่สมบูรณ์ โดยแต่ละองค์ประกอบสามารถเชื่อมต่อกันด้วยบัสหนึ่งหรือสอง ทิศทาง (Unidirectional or Bidirectional Bus) ส่วนกระบวนการในการควบคุมการเคลื่อนย้ายข้อมูลระหว่าง รีจิสเตอร์และวงจรลอจิกจะขึ้นอยู่กับพฤติกรรมของระบบที่กำหนดไว้ดังรูปที่ 2.16



รูปที่ 2.16 การออกแบบระบบเส้นทางของข้อมูล

ขั้นตอนถัดมาเป็นการออกแบบวงจรลอจิก ซึ่งจะเกี่ยวข้องกับการนำเกทดิจิตอลพื้นฐานและฟลิปฟลอป (filp flop) มาประกอบเป็นอุปกรณ์ย่อยต่างๆ เช่น รีจิสเตอร์เก็บข้อมูล บัส วงจรลอจิก และส่วนควบคุมฮาร์ดแวร์ ซึ่งผลลัพธ์ที่ได้ในขั้นตอนนี้จะเป็นเครือข่ายของการโยงใยระหว่างเกทและฟลิฟลอปนั้นเองการออกแบบในขั้นตอนถัดไป เป็นการเปลี่ยนเครือข่ายการโยงใยที่ได้จากขั้นตอนที่แล้วให้เป็นลำดับของทรานซิสเตอร์ (Transistor List) และ Layout ซึ่งขั้นตอนนี้จะเกี่ยวข้องโดยตรงกับการจัดวางทรานซิสเตอร์หรือไลบารีเซลล์เพื่อ แทนเกทและฟลิปฟลอปต่างๆ และในขั้นตอนสุดท้ายจะเป็นการส่งระบบที่ออกแบบไว้ไปทำการเจือสารที่โรงงานเพื่อผลิตออกมาเป็น วงจรรวมในที่สุด

2.3.2 ประวัติความเป็นมาของภาษา VHDL

VHDL ย่อมาจากคำว่า VHSIC Hardware Description Language (VHSIC : Very High Speed Integrated Circuit) เป็นภาษาโปรแกรมระดับสูง (High Level Language) ที่ใช้สำหรับการ

ออกแบบฮาร์ด แวร์ ในระบบดิจิตอล ตัวของภาษาสามารถบรรยายพฤติกรรมการทำงานในรูปของลำดับชั้น (Hierarchy) และสามารถเขียนได้หลายรูปแบบ ด้วยเหตุผลนี้จึงทำให้ภาษา VHDL เป็นเครื่องมือที่ใช้ออกแบบตั้งแต่ขั้นตอนบนสุด คือ แนวความคิดที่จะแก้ปัญหา ลงไปทีละขั้นจนถึงขั้นตอนของการสร้างวงจรจริง และตัวภาษาก็เปิดโอกาสให้วิศวกร ได้พัฒนาและจำลองการทำงานของรูปแบบฟังก์ชันการทำงานของวงจรอย่างสังเขป โดยยังไม่ต้องคำนึงถึงรายละเอียดเกี่ยวกับโครงสร้างวงจรจริง นอกจากนั้น VHDL ยังเป็นภาษาที่สนับสนุนลักษณะต่างๆ ของระบบดิจิตอลที่มีความซับซ้อนได้ทั้งหมด ดังนั้น VHDL จึงเป็นภาษาที่น่าสนใจในการศึกษาและนำไปใช้งานเป็นอย่างยิ่ง วิวัฒนาการของภาษา VHDL เริ่มต้นประมาณปี ค.ศ. 1981 เมื่อกระทรวงกลาโหมสหรัฐอเมริกาหรือ DoD (Department of Defense) ได้พยายามปรับปรุงอุปกรณ์อิเล็กทรอนิกส์และคอมพิวเตอร์ที่ใช้ในกิจการทางทหาร ให้มีความทันสมัยมากขึ้น ประกอบกับเทคโนโลยีทางด้านไมโครอิเล็กทรอนิกส์มีการพัฒนาไปอย่างรวดเร็วดังจะเห็นได้ จากการนำวงจรดิจิตอลหลายๆ วงจรมาทำการผลิตอยู่บนแต่ละซิลิคอนที่มีพื้นที่เพียง 1 – 2 ตารางเซนติเมตรเท่านั้น ซึ่งเป็นผลให้ประสิทธิภาพในการทำงานของวงจรสูงขึ้นตลอดจนความน่าเชื่อถือ ในการทำงานและความคงทนต่อสภาพแวดล้อมสูง แต่เนื่องจากในขณะนั้นขั้นตอนของการออกแบบ การผลิต และการตรวจสอบวงจรต้นแบบ เป็นขบวนการที่ต้องใช้วิศวกรและเวลาในดำเนินการมาก ฉะนั้นทาง DoD จึงจัดตั้งโครงการขึ้นมาเพื่อศึกษาวิธีการที่ช่วยในการพัฒนา วงจรอิเล็กทรอนิกส์ โดยเฉพาะอย่างยิ่งวงจรระบบดิจิตอล ให้สามารถนำไปผลิตได้เร็วขึ้น ซึ่งโครงการดังกล่าวมีชื่อว่า "Very High Speed Integrated Circuits" หรือ VHSIC โดยในระยะแรกนั้นโครงการนี้ถือเป็นความลับทางด้านความมั่นคงของประเทศ และอยู่ภายใต้ความควบคุมดูแลของ United States International Traffic and Arms Regulations (ITAR) สำหรับมาตรฐานของภาษาที่ใช้บรรยาย พฤติกรรมวงจรหรือฮาร์ดแวร์ของระบบสำหรับโครงการ VHSIC ที่ DoD ได้ให้ไว้สามารถสรุปได้ดังนี้

- ต้องเป็นภาษาที่นำไปเขียนรูปแบบระบบดิจิตอล และมีคุณสมบัติที่สามารถเข้าใจได้ทั้งมนุษย์และเครื่อง คอมพิวเตอร์โดยไม่ต้องมีการแปลหรือเปลี่ยนแปลงอีก
- สามารถนำไปใช้เป็นเอกสารประกอบโครงการได้
- ต้องเป็นภาษาที่เขียนขึ้นสำหรับใช้จำลองการทำงานของวงจร

ฉะนั้นภาษาดังกล่าวนี้จึงจัดเป็นภาษาโปรแกรมระดับสูง เช่นเดียวกับภาษาปาสคาล หรือภาษาซี ซึ่งในทางวิศวกรรม ภาษาที่ใช้ในการออกแบบฮาร์ดแวร์นี้เรียกว่า "Hardware Description Language" หรือ HDL ในตอนเริ่มแรกนั้น DoD ได้มอบหมายให้บริษัทไอบีเอ็ม เท็กซัสอินสตูเมนท์ และอินเตอร์เมทริกซ์ เป็นผู้ศึกษาและพัฒนาโครงการ ซึ่งการดำเนินงานเป็นไปอย่างต่อเนื่องจนกระทั่งในปี ค.ศ. 1985 ทาง ITAR ได้ยกเลิกข้อจำกัดในการถ่ายทอดเทคโนโลยีทางทหารออก

จากโครงการนี้ ดังนั้นภาษา VHDL จึงเริ่มเป็นที่รู้จักกันโดยทั่วไป และประมาณปี ค.ศ. 1987 IEEE ได้ทำการกำหนดมาตรฐานของภาษานี้เป็น IEEE 1076-1987 และมีชื่อเรียกว่า VHDL ซึ่งมาตรฐานนี้ได้รับ การปรับปรุงจนเป็นมาตรฐาน IEEE 1076-1993 หรือ VHDL 1993 เนื่องจากในขณะนั้น DoD เป็นลูกค้ารายใหญ่ ของอุตสาหกรรมอิเล็กทรอนิกส์และคอมพิวเตอร์ ดังนั้นจึงมีผู้รับโครงการต่างๆ จาก DoD ไปดำเนินการวิจัยและพัฒนา เป็นจำนวนมาก และเพื่อให้ทุกโครงการอยู่ในมาตรฐานเดียวกันหมด ดังนั้นทาง DoD จึงได้กำหนดว่า ทุกๆ โครงการต้อง เขียนอยู่ในรูปของภาษา VHDL เท่านั้น ซึ่งทำให้ DoD สามารถนำโครงการเหล่านี้ไปจำลองกับเครื่องคอมพิวเตอร์ได้หลายๆ ระบบ

2.3.3 ข้อกำหนด

DoD ได้ตั้งข้อกำหนดสำหรับภาษา VHDL ในเดือนมกราคมปี ค.ศ. 1983 ไว้ดังนี้

2.3.3.1 ลักษณะทั่วไป

DoD ได้กำหนดให้ VHDL เป็นภาษาสำหรับการออกแบบและบรรยายของฮาร์ดแวร์ ซึ่งหมายถึง ความสามรถในการอธิบายและออกแบบในระดับสูง การจำลอง (Simulation) การสังเคราะห์ (Synthesis) และการทดสอบ (Testing) นอกจากนั้น VHDL ยังถูกกำหนดไว้สำหรับการบรรยายฮาร์ดแวร์ตั้งแต่ระดับบนซึ่งก็คือระบบจนถึง ระดับเกทอีกด้วย เนื่องจากในการทำงานของระบบดิจิตอลนั้น ทุกๆ องค์ประกอบภายในระบบไม่ว่าเล็กหรือใหญ่ จะทำงานไปพร้อมๆ กัน ซึ่งในเรื่องของความพร้อมเพรียงในการทำงานนี้ก็ถือเป็นข้อกำหนดที่สำคัญอย่างหนึ่งของ VHDL ด้วยเช่นกัน (สำหรับในภาษาที่ใช้ในการบรรยายฮาร์ดแวร์นั้นความพร้อมเพรียงจะหมายถึงทุกๆ คำสั่ง องค์ประกอบ เกทหรือวงจรลอจิกจะถูกนำมาปฏิบัติทั้งหมด ดังนั้นในที่สุดแล้วก็จะดูเหมือนว่าได้มีการปฏิบัติไปพร้อมๆ กัน)

2.3.3.2 สนับสนุนการออกแบบแบบลำดับขั้น

การออกแบบแบบลำดับขั้นเป็นลักษณะที่สำคัญอย่างหนึ่งสำหรับออกแบบระบบที่มีหลายๆ ระดับ โดยในการ ออกแบบจะประกอบด้วยส่วนการบรรยายการเชื่อมต่อ และส่วนการบรรยายห้าที่การทำงาน ซึ่งหน้าที่การทำงาน ของระบบสามารถกำหนดได้ด้วยตัวเอง หรืออาจถูกกำหนดโดยโครงสร้างที่ประกอบด้วยองค์ประกอบย่อยๆ ลงไปได้เช่นกัน แต่ที่ระดับล่างสุดองค์ประกอบต้องถูกบรรยายหน้าที่การทำงานด้วยตัวมันเอง และไม่สามารถกำหนด การทำงานโดยลักษณะแบบโครงสร้างได้

2.3.3.3 ไลบรารี

VHDL ได้สนับสนุนการมีไลบรารีเพื่อระบบการจัดการที่ดี ผู้ออกแบบสามารถกำหนดลักษณะและการทำงานของ อุปกรณ์พื้นฐานไว้ในระบบไลบรารี หรือจะใช้ไลบรารีที่ระบบ

ได้จัดเตรียมไว้แล้วก็ได้ โมเดลและการบรรยายที่ถูกต้องควรจัดเก็บไว้ไลบรารีหลังจากที่ได้ผ่านการคอมไพล์เรียบร้อยแล้วเพื่อให้ผู้ออกแบบคนอื่นๆ สามารถนำไปใช้ได้ด้วย

2.3.3.4 ลำดับคำสั่ง

แม้ว่าการปฏิบัติคำสั่งหรือกระบวนการโดยพร้อมเพรียงกันจะเป็นคุณสมบัติที่สำคัญของ VHDL ก็ตาม ตัวภาษาเองก็ยังมีการจัดเตรียมลักษณะการควบคุมแบบลำดับคำสั่งไว้ให้ด้วย เมื่อผู้ออกแบบได้กำหนดหน้าที่และองค์ประกอบ ที่ทำงานพร้อมกันของระบบไว้เรียบร้อยแล้ว ผู้ออกแบบยังสามารถบรรยายหน้าที่การทำงานซึ่งเป็นรายละเอียดภายใน ของแต่ละองค์ประกอบได้ในลักษณะเดียวกับการเขียนโปรแกรมที่ประกอบด้วยโครงสร้างแบบ case, if – then – else และ loop ทั่วๆ ไปได้ การบรรยายแบบลำดับคำสั่งทำให้การออกแบบหน้าที่การทำงานของอุปกรณ์กระทำได้ สะดวกและง่ายขึ้น อย่างไรก็ตามโครงสร้างทั้งหมดของ VHDL ก็ยังคงเป็นการทำงานแบบพร้อมเพรียงกันเช่นเดิม

2.3.3.5 การกำหนดคุณสมบัติ

นอกจากการกำหนดอินพุทและเอาท์พุทแล้ว เงื่อนไขอื่นๆ ก็มีผลต่อการปฏิบัติหน้าที่ของอุปกรณ์ฮาร์ดแวร์ด้วยเช่นกัน โดยสิ่งนี้รวมถึงสภาพแวดล้อมและลักษณะทางกายภาพของอุปกรณ์นั้นๆ ด้วย ซึ่งภาษาสำหรับการออกแบบที่ดีควร ให้ผู้ออกแบบกำหนดคุณสมบัติของอุปกรณ์ที่ใช้ได้ด้วย เช่น สามารถกำหนดขนาด ลักษณะทางกายภาพเวลา โหลด และเงื่อนไขทางสภาพแวดล้อมอื่นๆ ซึ่งความสามารถในการกำหนดคุณสมบัตินี้ก็เป็นส่วนหนึ่งที่มีอยู่ในภาษา VHDL ด้วยเช่นกัน

2.3.3.6 ชนิดของข้อมูล

VHDL สามารถกำหนดชนิดของข้อมูลไม่เพียงแต่ชนิด BIT และ BOOLEAN เท่านั้น แต่ยังสามารถกำหนดชนิด ของข้อมูลเป็นจำนวนเต็ม จำนวนจริง จุดทศนิยม และชนิดลำดับการนับ (Enumerate Type) หรือแม้แต่ชนิดของ ข้อมูลที่ผู้ออกแบบกำหนดขึ้นมาเองก็ได้

2.3.3.7 โปรแกรมย่อย

ความสามารถในการใช้ฟังก์ชั่นและโพรซีเจอร์ (Procedure) ก็เป็นข้อกำหนดอีกอย่างหนึ่งใน VHDL ซึ่งผู้ออกแบบ สามารถนำโปรแกรมย่อยมาใช้ในการเปลี่ยนแปลงชนิดของข้อมูล การกำหนดหน่วยของลอจิก การกำหนดตัวกระทำต่างๆ หรือหน้าที่อื่นๆ ตามที่ต้องการได้เช่นเดียวกับการเขียนโปรแกรมทั่วไป

2.3.3.8 การควบคุมเวลา

VHDL อนุญาตให้ผู้ออกแบบสามารถกำหนดเวลาในการส่งผ่านข้อมูลหรือสัญญาณได้ตามต้องการการตรวจสอบ การออกแบบเกทหรือการหน่วงเวลาก็สามารถกระทำได้โดย

การกำหนดช่วงเวลาที่แน่นอนหรือ กำหนดให้มีการรอคอยเหตุการณ์ (Event) นอกจากนี้ก็ยังสามารถกำหนดรูปแบบของสัญญาณนาฬิกาได้อีกด้วย

2.3.3.9 การกำหนดแบบโครงสร้าง

การกำหนดโครงสร้างขององค์ประกอบต่างๆ สามารถกระทำได้ในทุกระดับของการออกแบบโดยการกำหนดโครงสร้างขององค์ประกอบร่วมที่เกิดจากองค์ประกอบย่อยซึ่งแตกต่างกันหรือ เหมือนกันก็เป็นข้อกำหนดอย่างหนึ่งของ VHDL เช่นกัน

2.3.4 องค์ประกอบพื้นฐานของ VHDL

รูปแบบพื้นฐานที่ใช้ในการบรรยายถึงองค์ประกอบของ VHDL จะประกอบไปด้วยส่วนกำหนดการเชื่อมต่อ (Interface) และส่วนกำหนดลักษณะเชิงสถาปัตยกรรม (Architecture) ดังแสดงในรูปที่ 3.16 โดยในการบรรยายการเชื่อมต่อจะขึ้นต้นด้วยคำว่า ENTITY แล้วตามด้วยชื่อขององค์ประกอบจากนั้นตามด้วยคำว่า IS และถัดมาจะเป็นการบรรยายถึงพอร์ต การติดต่อ อินพุท – เอาท์พุท ขององค์ประกอบ ส่วนลักษณะภายนอกอื่นๆ เช่น เวลา อุณหภูมิก็สามารถรวมเข้าไปในส่วนนี้ ได้เช่นกัน ในส่วนของการกำหนดลักษณะเชิงสถาปัตยกรรมจะขึ้นต้นด้วยคำว่า ARCHITECTURE ซึ่งเป็นส่วนที่ใช้บรรยายหน้าที่การทำงานขององค์ประกอบ โดยหน้าที่การทำงานนี้จะขึ้นอยู่กับสัญญาณอินพุท เอาท์พุทและพารามิเตอร์อื่นๆ ที่ได้กำหนดไว้ในส่วนของการเชื่อมต่อดังรูปที่ 2.17 และสำหรับการบรรยายหน้าที่ขององค์ประกอบจะเริ่มต้นหลังจากคำว่า BEGIN เป็นต้นไป

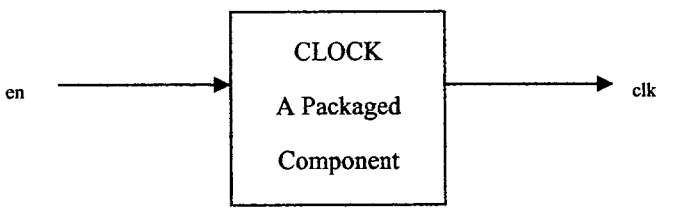
```
ENTITY component_name IS
        Input and output parts
        Physical and other parameters
                END [component_name] ;:


ARCHITECTURE identifier OF component_name IS
        [declaration]
BEGIN
```

รูปที่ 2.17 การกำหนดการเชื่อมต่อและสถาปัตยกรรม

2.3.4.1 การกำหนดการเชื่อมต่อ

การกำหนดการเชื่อมต่อเป็นระดับบนสุดของการออกแบบ โดยในระดับนี้ต้องกำหนดพอร์ตสำหรับการติดต่อกับองค์ประกอบ ภายนอกอื่นๆ ดังตัวอย่างในรูปที่ 2.18 ซึ่งเป็นบล็อกไดอะแกรม และการบรรยายการเชื่อมต่อขององค์ประกอบสำหรับตัวจ่าย สัญญาณนาฬิกา ในบรรทัดแรกของการบรรยายการเชื่อมต่อเป็นการกำหนดชื่อขององค์ประกอบซึ่งกำหนดเป็น clock_componenet ตามด้วยคำว่า PORT และชื่อของพอร์ตอยู่ภายในวงเล็บ ส่วน IN และ OUT เป็นการกำหนด โหนดของสัญญาณให้เป็นอินพุทหรือเอาท์พุท และ BIT เป็นการแสดงชนิดของข้อมูล




```
ENTITY clock_component IS
PORT (en : IN BIT ; ck : OUT BIT)
END clock _name;
```

รูปที่ 2.18 บล็อคไดอะแกรมและการบรรยายการเชื่อมต่อของ clock_component

2.3.4.2 การกำหนดรูปแบบการบรรยาย

หน้าที่การทำงานขององค์ประกอบจะถูกบรรยายในส่วนนี้ ซึ่งในการบรรยายสามารถกำหนดค่าของสัญญาณ เอาท์พุทในเทอมของอินพุทหรือในรูปขององค์ประกอบอื่นๆ หรือทั้งสองอย่างรวมกันก็ได้ ดังตัวอย่างการบรรยายของ clock_component ในรูปที่ 2.19 ซึ่งเป็นการบรรยายในเชิงพฤติกรรมโดยมี en เป็นอินพุทและ ck เป็นเอาท์พุท PROCESS เป็นคำที่ใช้ในการเริ่มต้นสำหรับการบรรยายในเชิงพฤติกรรมและภายในโปรเซสกำหนดให้ periodic เป็นตัวแปรที่มีคำเริ่มต้นเป็น "0" ถ้าสัญญาณ en มีค่าเป็น "1" จะทำให้ตัวแปร periodic ถูกคอมพลีเมนท์ (complement) และส่งค่าให้กับ clk ซึ่งเป็นสัญญาณเอาท์พุท และสำหรับคำสั่ง WAIT จะเป็นการกำหนดให้สัญญาณมีคาบเวลาเท่ากัน 1 ไมโครวินาที

```
ARCHITECTURE behavional OF clock_component IS
BEGIN
        PROCESS
                VARIABLE periodic : BIT := '0'
        BEGIN
                IF en = '1' THEN
                Periodic := Not periodic;
                END IF;
                Ck<=periodic;
                WAIT FOR 1 US;
END PROCESS; END behavioral;
```

รูปที่ 2.19 การบรรยายเชิงพฤติกรรมของ clock_component

2.3.5 หน่วยการออกแบบแพ็กเก็ต

ข้อมูลต่างๆ ตลอดจนโปรแกรมย่อย ที่เป็นประโยชน์ต่อการเขียนรูปแบบการบรรยายระบบดิจิตอล สามารถเก็บไว้ในส่วนของแพ็กเก็ต ซึ่งหน่วยการออกแบบต่างๆ เช่น หน่วยการออกแบบ Entity หน่วยการออกแบบสถาปัตยกรรมหรือหน่วยการออกแบบแพ็กเกจอื่นๆ สามารถเรียกข้อมูลเหล่านี้ไปใช้ได้ นอกจากนั้นสิ่งที่นิยมทำกันมากคือการนำรูปแบบมาตรฐานต่างๆ เช่น อุปกรณ์มาตรฐาน (เช่น ไอซีตระกูล 74XX เป็นต้น) มาเก็บไว้ในรูปของแพ็กเกจที่ทุกคนสามารถเข้าถึงได้ ตามปกติแล้วแพ็กเกจจะแบ่งออกเป็น 2 ส่วนคือ การประกาศแพ็กเกจ(Packge declaration) และส่วนของบอดี้แพ็กเกจ (Package body) เนื่องจาก แพ็กเกจถูกสร้างขึ้นเป็นส่วนแยกต่างหากออกจากรูปแบบที่กำลังเขียนอยู่ ฉะนั้นการที่นำแพ็กเกจไปใช้นั้นจะต้องมีการเชื่อมโยงหรืออ้างอิงเสียก่อน ซึ่งในภาษา VHDL สามารถกระทำได้ด้วยชุดคำสั่ง USE

2.3.5.1 PACKGE DECLARATION

ส่วนที่มีความสำคัญที่สุดของแพ็กเกจ (ถ้ามองในแง่ของการนำไปใช้จากภายนอก) ได้แก่ส่วนการประกาศแพ็กเกจ เนื่องจากเป็นส่วนที่ใช้กำหนดชื่อของสิ่งที่ประกาศอยู่ภายในแพ็กเกจ สำหรับนำไปใช้ภายนอกตัวของแพ็กเกจเอง ถ้ามี การประกาศสิ่งใดๆ ในส่วนขอส่วนบอดี้แพ็กเกจ แต่ไม่ถูกประกาศในส่วนการประกาศแพ็กเกจจะทำให้ค่าและพฤติกรรมไม่สามารถนำไปใช้งานในส่วนนอกได้ซึ่งเปรียบเทียบได้กับสิ่งที่ประกาศไว้ในส่วนของการประกาศ Entity คือ จุดเชื่อมต่อ หรือ พอร์ต ที่มีหน้าที่ติดต่อกับโลกภายนอก ฉะนั้นโดยทั่วไปแล้วแพ็กเกจสามารถสร้างขึ้นได้โดยไม่จำเป็น ต้องมีส่วนบอดี้ และยังสามารถนำไปใช้งานจากรูปแบบภายนอกได้เช่น

ใช้สำหรับประกาศ ชนิด (Type) หรือสัญญาณ เช่นเดียวกับ ส่วนบอดี้แพ็กเก็จที่ไม่จำเป็นต้องมีส่วนของการประกาศแพ็กเกจ แต่แพ็กเกจนั้นจะไม่สามารถนำไปใช้จากรูปแบบอื่นได้

```
PACKAGE package_name IS
        Package_declarative_part
END package_name;
```

รูปที่ 2.20 โครงสร้างทั่วไปของส่วนการประกาศแพ็กเกจ

2.3.5.2 PACKGE BODY

โครงสร้างซึ่งประกอบด้วยลำดับคำสั่งที่ใช้บรรยายฟังก์ชันการทำงานของโปรแกรมย่อยทั้งหลาย ซึ่งชื่อของโปรแกรมย่อยนั้นๆ ได้ถูกประกาศไปแล้วในส่วนของการประกาศแพ็กเก็จ จะถูกเก็บไว้ในส่วนของบอดี้แพ็กเกจ ทั้งนี้รวมถึงการกำหนดค่าคงที่ต่างๆ อันได้แก่ค่าคงที่ที่ถูกประกาศชื่อไว้ก่อนในส่วนของการประกาศแพ็กเกจ และถูกกำหนดค่าในส่วนของบอดี้แพ็กเกจ ฉะนั้นในส่วนของบอดี้แพ็กเกจจึงไม่จำเป็นต้องมี ถ้าในส่วนของการประกาศแพ็กเกจไม่มีการ ประกาศชื่อที่เป็นโปรแกรมย่อย หรือค่าคงที่ การเขียนบอดี้แพ็กเกจนั้นจะเป็นไปตามกฎเกณฑ์ดังแสดงในรูป ที่ 2.21

```
PACKAGE BODY package_name IS
        declarative_part
END package_name;
```

รูปที่ 2.21 โครงสร้างของบอดี้แพ็กเกจ

2.3.6 หน่วยการออกแบบ Configuration

ดังที่ทราบกันแล้วว่าระบบดิจิตอลรูปแบบหนึ่งไม่ว่าจะเป็นอะไรก็ตาม จะสามารถมีหน่วยการออกแบบ Entity ได้ เพียงหนึ่งเดียวเท่านั้น ซึ่งในหน่วยการออกแบบ Entity หนึ่งหน่วยนี้อาจจะมีสถาปัตยกรรมที่เป็นหน่วยรองได้หลายหน่วย ดังนั้นจะต้องมีหน่วยการออกแบบ Configuration มาเพื่อกำหนดการใช้ Configuration ของการประกอบ Entity กับหน่วยการออกแบบสถาปัตยกรรมหน่วยใดๆ เข้าด้วยกัน

```
COMFIGURTRION identifier OF entity name IS
        Configuration_declarative_part
```

รูปที่ 2.22 โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ

2.3.7 โปรแกรมย่อย

การใช้ฟังก์ชันและโพรซีเจอร์ใน VHDL เปรียบได้กับการใช้โปรแกรมย่อยในการเขียนโปรแกรมภาษาชั้นสูงทั่วๆ ไปค่าที่ถูกส่งกลับหรือถูกเปลี่ยนแปลงโดยโปรแกรมย่อยอาจจะมีหรือไม่มีผลต่อฮาร์ดแวร์โดยตรงก็ได้ เช่นถ้าใช้ฟังก์ชัน แทนการกระทำในสมการบูลีนก็จะมีผลต่อวงจรลอจิกจริงๆ ในขณะที่ถ้าใช้โปรแกรมย่อยในการเปลี่ยนชนิดของข้อมูล หรือในการคำนวณค่าการหน่วงเวลาแล้วก็จะไม่มีผลต่อโครงสร้างของฮาร์ดแวร์ รูปที่ 2.23 แสดงการใช้โพรซีเจอร์ เพื่อเปลี่ยนข้อมูลชนิด 8 บิตเป็นค่าจำนวนเต็ม และรูปที่ 2.24 แสดงการใช้ฟังก์ชันโดยกำหนดให้ X เป็นตัวแปรชนิด บิตแทนการกระทำในสมการบูลีน

```
TYPE byte IS ARRAY ( 7 DOWNTO 0 ) OF BIT:
...
PROCEDURE byte_to_integer(ib : IN byte; oi : OUT INTEGER) IS
        VARIABLE result : INTEGER := 0;
BEGIN
        FOR I IN 0 TO 7 LOOP
                IF ib(i) = '1' THEN
                result := result + 2**I;
```

รูปที่ 2.23 การใช้โพรซีเจอร์

```
FUNCTION f (a , b, c:BIT ) RETURN BIT IS
        VARIABLE x: BIT;
BEGIN
        x := ((NOT a ) AND (NOT b) AND c);
```

รูปที่ 2.24 การใช้ฟังก์ชั่น

2.3.8 โอเปอร์เรเตอร์

การบรรยายเชิงพฤติกรรมในภาษา VHDL มีตัวดำเนินการหรือโอเปอร์เรเตอร์ทางลอจิกและคณิตศาสตร์เช่นเดียว กับภาษาซอฟต์แวร์ทั่วไป

PREDEFIND OPERATORS
LOGICAL OPERATORS : NOT AND OR NAND NOR X OR OPERAND TYPE : BIT BOOLEAN RESULT TYPE : BIT BOOLEAN
RELATIONAL OPERATORS : =/=<< = >> OPERAND TYPE : any type RESULT TYPE : Boolen
ARITHMETIC OPERATORS : + - * / ** MOD REM ABS OPERAND TYPE : INTEGER REAL Physical RESULT TYPE : INTEGER REAL Physical

รูปที่ 2.25 ตัวดำเนินการใน VHDL

2.3.9 เวลาและความพร้อมเพรียง

ในวงจรอิเล็กทรอนิกส์อุปกรณ์ทุกๆ ตัวจะอยู่ในสภาพเตรียมพร้อมเสมอ (Always Active) และจะมีเรื่องของเวลา เข้ามาเกี่ยวข้องในทุกๆ เหตุการณ์ที่เกิดขึ้นเสมอ VHDL เป็นภาษาที่ได้รับการออกแบบมาเพื่อให้สามารถบรรยายรูป แบบและการพ้องกันของเวลาสำหรับการทำงานของอุปกรณ์ได้อย่างถูกต้อง การบรรยายการทำงานที่อยู่ภายในส่วน ของการบรรยายสถาปัตยกรรม จะมีการทำงานที่พร้อมเพรียงกันเสมอ หรือแม้แต่โปรเซสซึ่งมีการทำงานภายในเป็นแบบลำดับคำสั่งก็ตาม ซึ่งหากมีหลายๆ โปรเซสอยู่ภายในโครงสร้างเดียวกันทุกๆ โปรเซสก็จะทำงานไปพร้อมๆ กันด้วย

2.3.10 สัญญาณและตัวแปร

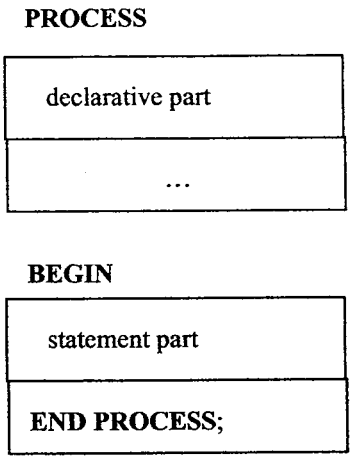
สัญญาณมีลักษณะเป็นเสมือนตัวกลางฮาร์ดแวร์ที่ใช้ในการส่งผ่านข้อมูลและมีเรื่องของเวลาเข้ามาเกี่ยวข้องด้วยการ กำหนดค่าให้กับสัญญาณจะใช้สัญลักษณ์ <= ในการส่งค่าและสามารถใช้คำสั่ง AFTER เพื่อกำหนดช่วงเวลาในการ ส่งผ่านค่าของสัญญาณ เช่น w>=a AFTER 12 NS หมายถึงการกำหนดค่าสัญญาณ a ให้กับ w หลังจากเวลา ผ่านไป 12 นาโนวินาที ในทางตรงข้ามตัวแปรมีลักษณะเป็นเสมือนตัวกลางที่ใช้ในการส่งผ่านข้อมูลและไม่มีเรื่องของเวลาเข้ามาเกี่ยวข้องด้วย ซึ่งตัวแปรจะถูกใช้ในส่วนที่มีการทำงานเป็นแบบลำดับคำสั่งเช่นใน ฟังก์ชัน โพรซีเจอร์ และโปรเซส สำหรับการกำหนดค่าให้กับตัวแปรจะใช้สัญลักษณ์ " = "

2.3.11 การบรรยายเชิงพฤติกรรม

การบรรยายลักษณะการทำงานของอุปกรณ์ฮาร์ดแวร์ในเชิงพฤติกรรม เป็นการบรรยายลักษณะการเปลี่ยนแปลงของข้อมูลในรูปแบบของอัลกอริธึมสำหรับการคำนวณผลลัพธ์ที่เกิดขึ้น ซึ่งสืบเนื่องมาจากการเปลี่ยนแปลงสภาวะของข้อมูล ที่เข้ามาโดยไม่คำนึงถึงลักษณะโครงสร้างหรือความสัมพันธ์ของอุปกรณ์ที่อยู่ภายในว่าจะเป็นอย่างไร ในหัวข้อนี้จะแสดงถึงการบรรยายเชิงพฤติกรรม แทนการใช้โมดูลฮาร์ดแวร์รวมถึงข้อกำหนดต่างๆ ที่ควรรู้

2.3.12 โปรเซส

โปรเซสเป็นรูปแบบพื้นฐานอย่างหนึ่งที่ใช้ในการกำหนดให้กับสัญญาณ โปรเซสจะอยู่ในสถานะที่เตรียมพร้อมอยู่เสมอ และจะปฏิบัติคำสั่งพร้อมๆ กันกับโปรเซสอื่นๆ ที่อยู่ในสถาปัตยกรรมบรรยายเดียวกัน โดยโปรเซสจะปฏิบัติงานตามคำสั่งทันทีที่มีเหตุการณ์เกิดขึ้นกับสัญญาณที่อยู่ทางด้านขวามือของสัญลักษณ์กำหนดค่าให้กับสัญญาณ (<=) การบรรยาย โปรเซสจะเริ่มต้นด้วยคำสั่ง PROCESS และจบด้วยคำสั่ง END PROCESS ในรูปที่ 2.26 เป็นการแสดงส่วนประกอบของการบรรยายแบบโปรเซส ซึ่งประกอบด้วยส่วนของการประกาศตัวแปรที่ต้องใช้ ส่วนของการปฏิบัติคำสั่งเพื่อให้ได้ผลลัพธ์ที่ต้องการ



รูปที่ 2.26 รูปแบบของการบรรยายแบบโปรเซส

2.3.13 การกำหนดตัวดำเนินการภายในโปรเซส

ตัวดำเนินการภายในโปรเซสมี 3 ชนิดคือ ตัวแปร (Variable) ไฟล์ (File) และตัวคงที่ (Constant) ซึ่งตัวดำเนินการทั้งสามชนิดนี้หากมีการประกาศไว้ในโปรเซสใดก็จะใช้ได้เฉพาะภายในโปรเซสนั้นเท่านั้นสำหรับการติดต่อกับภายนอกหรือระหว่างโปรเซสสามารถทำได้โดยใช้สัญญาณ (Signal) หรือตัวคงที่ที่ได้ประกาศไว้ในส่วนของ ARCHITECTURE ในรูปที่ 2.27 แสดงตัวอย่างการประกาศตัวกระทำภายในโปรเซสซึ่งจะอยู่ระหว่างคำสั่ง PROCESS และ BEGIN และค่าเริ่มต้นที่ถูกกำหนดให้กับตัวดำเนินการภายในโปรเซสจะถูกนำมาใช้ในตอนเริ่มต้น ของการปฏิบัติเพียงครั้งเดียวเท่านั้น ต่างกับค่าเริ่มต้นที่อยู่ภายในโปรแกรมย่อยจะถูกนำมาใช้ทุกครั้งที่มีการเรียกใช้โปรแกรมย่อยนั้นๆ

```
PROCESS
        FILE flush : TEXT IS IN "filename.dat";
VARIABLE var : BIT;
        CONSTANT n : INTEGER := 0;
BEGIN
```

รูปที่ 2.27 ตัวอย่างการประกาศตัวดำเนินการภายในโปรเซส

2.3.14 การกำหนดการกระทำภายในโปรเซส

การกระทำใดๆ ภายในโปรเซสจะเป็นการปฏิบัติแบบลำดับ (Sequential) เสมอ ซึ่งภายในโปรเซสสามารถใช้ประโยคเงื่อนไขหรือการทำซ้ำได้เช่น IF-THEN – ELSE, CASE – WHEN , FOR LOOP และ WHILE LOOP ดังตัวอย่างในรูปที่ 2.28 และ 2.29

```
ARCHITECTURE demo OF partial_process IS
...
BEGIN
        PROCESS
        ...
        BEGIN
                ...
                x <= '1';
                IF x = '1' THEN
                        perform action_1
                ELSE
```

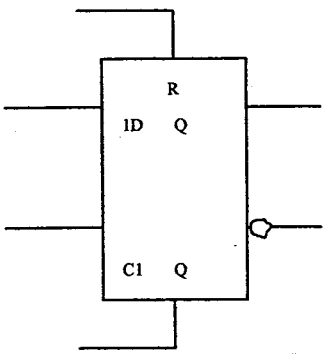
รูปที่ 2.28 เงื่อนไขการกระทำในโปรเซส

```
ARCHITECTURE demo OF partial_process IS
  ...
BEGIN
    PROCESS
BEGIN
                ...
                x <= a AFTER 10 NS;
                y <= b AFTER 6 NS;
```

รูปที่ 2.29 แสดงการกระทำในโปรเซส

2.3.15 การกระตุ้นและยับยั้งการกระทำของโปรเซส

การกระทำภายในโปรเซสจะอยู่ในสภาพเตรียมพร้อม และมีการปฏิบัติงานอยู่ตลอดเวลาที่มีการเปลี่ยนแปลงของเหตุการณ์เกิดขึ้น อย่างไรก็ดีเราสามารถกระตุ้นหรือยับยั้งการกระทำภายในโปรเซสได้โดยการกำหนดรายการของสัญญาณที่ต้อง การให้โปรเซสปฏิบัติงานเมื่อมีเหตุการณ์เกิดขึ้นกับสัญญาณที่กำหนดไว้เท่านั้น ส่วนเหตุการณ์ใดๆ ที่เกิดขึ้นกับสัญญาณ ที่ไม่ได้กำหนดไว้ในรายการก็จะไม่ส่งผลให้มีการกระทำภายในโปรเซส ซึ่งรายการของสัญญาณนี้เรียกว่า Sensitivity List และจะกำหนดไว้ภายในวงเล็บหลังคำสั่ง PROCESS รูปที่ 2.30 (a) แสดงตัวอย่างโมเดล และรูปที่ 3.29 (b) เป็นตัวอย่างการบรรยายการเชื่อมต่อของ I-Filp Frop ส่วนรูปที่ 2.31 แสดงถึงการบรรยายเชิงพฤติกรรมของ D-Flip Flop โดยในรูปที่ 2.31 (a) เป็นการใช้ตัวกระจายภายนอกโปรเซส และรูปที่ 2.31 (b) เป็นการใช้ตัวกระทำภายในโปรเซส โดยมีรายการของสัญญาณ (rst, set. Clk) เป็นตัวกระตุ้นการปฏิบัติงานภายในโปรเซส



(a)

```
ENTITY companent_name IS
        Input and output ports
        Physical and other parameters
END component_name;
```

(b)

รูปที่ 2.30 (a) ตัวอย่างโมเดล D-Flip Flop, (b) การบรรยายการเชื่อมต่อของ D-Flip Flop

```
ARCHITECTURE behavioral OF d_sr_flipflop IS
        SIGNAL state : BIT := '0';
BEGIN
        dff : PEOCESS (rst,set,clk)
                BEGIN
           IF set = '1' THEN
                  state <= '1' AFTER sq_delay;
                ELSIF rst = '1' THEN
                  state <= '0' AFTER sq_delay;
                ELSIF clk = '1' AND clk 'EVENT THEN
                  state <= d AFTER sq_delay;
```

(a)

```
ARCHITECTURE average_daldy_behavioral OF d_sr_flipflop IS
BEGIN
        dff : PEOCESS (rst,set,clk)
                  VARIABLE state : BIT := '0';
                BEGIN
           IF set = '1' THEN
                  state <= '1';
                ELSIF rst = '1' THEN
                  state <= '0';
                ELSIF clk = '1' AND clk 'EVENT THEN
                  state <= d;
```

(b)

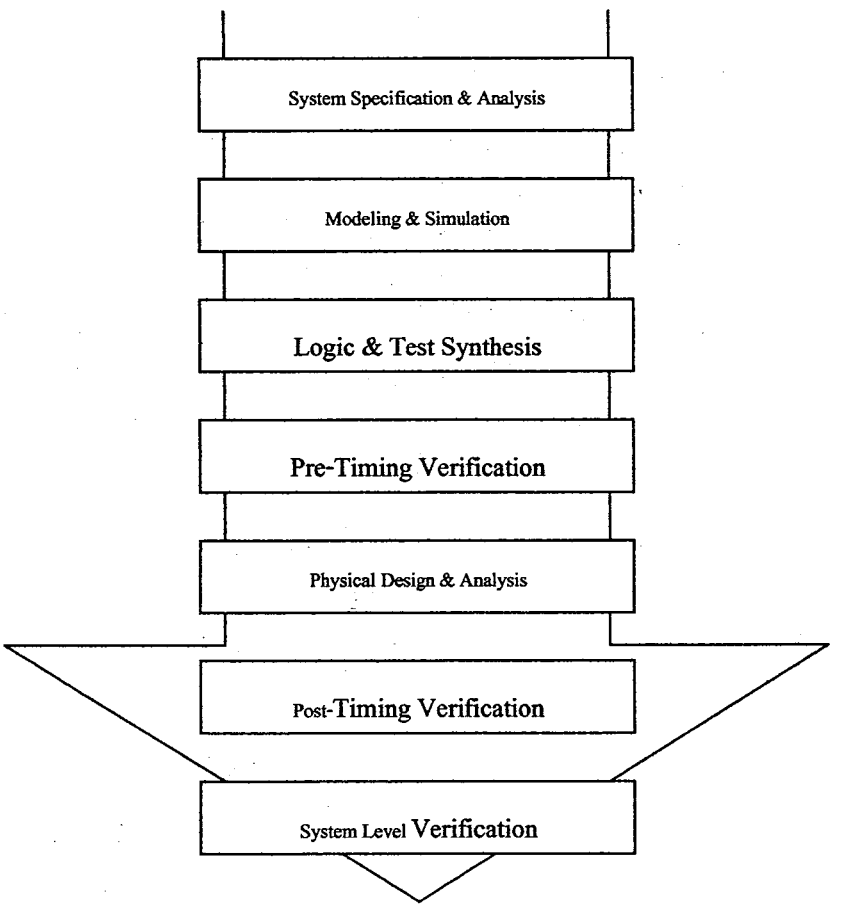
รูปที่ 2.31 การบรรยายเชิงพฤติกรรมของ D-FlipFlop

(a) การใช้ตัวกระทำภายนอกโปรเซส

(b) การใช้ตัวกระทำภายในโปรเซส

2.3.16 การออกแบบจากบนลงล่าง

ในการพัฒนาวงจรรวมดิจิตอลขนาดใหญ่ที่มีความซับซ้อน วิศวกรหรือผู้ออกแบบมักจะมองการออกแบบให้อยู่ในรูปของ บล็อกไดอะแกรมก่อนที่ทำวิเคราะห์ให้ลึกถึงรายละเอียดต่อไป ซึ่งภาษา VHDL นั้นอนุญาตให้อธิบายและวิเคราะห์การทำงานของแต่ละบล็อก รวมถึงการปรับปรุงการทำงานจากผลที่วิเคราะห์เพื่อให้ได้การทำงานตามต้องการ นอกจากนี้ยัง สามารถเพิ่มเติมในรายละเอียดในแต่ละขั้นตอนได้ซึ่งหลักการนี้สอดคล้องกับหลักการออกแบบจากบนลงล่าง (Top-Down Design) นั่นเอง ถ้าทดลองเปรียบเทียบกับการออกแบบจากล่างขึ้นบน (Bottom – Up Design) จะเห็นได้ว่า การออกแบบจากล่างขึ้นบนจะใช้เวลาการออกแบบมากกว่า 90% เนื่องจากเป็นการวาดวงจรด้วยอุปกรณ์ต่างๆ (Schematic capture) ที่ประกอบกันเข้าเป็นวงจรที่ต้องการออกแบบก่อนแล้วจึงทำการจำลองการทำงาน และตรวจ สอบความถูกต้อง ดังนั้นการใช้ภาษา VHDL กับหลักการออกแบบจากบนลงล่างจึงเป็นทางออกให้กับวิศวกรให้สามารถ ออกแบบและพัฒนาวงจรที่มีความซับซ้อนได้มากขึ้น ทั้งยังช่วยลดเวลาและค่าใช้จ่ายในการออกแบบด้วย



รูปที่ 2.32 ขั้นตอนการออกแบบจากบนลงล่าง

จากรูปที่ 2.32 แสดงถึงขั้นตอนของการออกแบบจากบนล่าง ทั้งนี้ให้ทางปฏิบัติอาจมีข้อแตกต่างไปจากนี้บ้าง เล็กน้อยเนื่องจากขั้นตอนของการผลิต (Implementation) สามารถกระทำได้หลายเทคโนโลยีสำหรับรายละเอียดของขั้นตอน การออกแบบจากบนล่างในแต่ละขั้นตอนมีดังนี้

2.3.16.1 สร้างข้อกำหนดของความต้องการ และวิเคราะห์ระบบเพื่อหาแนวความคิดและหลักการ (Idea and Concept) ใน การแก้ปัญหา

2.3.16.2 เขียนรูปแบบของระบบที่ต้องการออกแบบโดยใช้ภาษา VHDL หรือ ภาษา HDL อื่นๆ สำหรับบรรยายพฤติกรรมการ ทำงานพร้อมทั้งจำลองการทำงานเพื่อเปรียบเทียบและตรวจสอบความถูกต้องกับข้อกำหนด

2.3.16.3 หลังจากที่ได้หลักการขั้นต้นพร้อมแนวความคิดที่ผ่านการตรวจสอบแล้วหลักการนี้จะถูกเพิ่มเติมในรายละเอียดลงมา เป็นลำดับขั้นที่สอง จนกระทั้งอยู่ในระดับที่จะนำไปผลิตวงจริง หรือสังเคราะห์ในขั้นตอนนี้เองเทคโนโลยีที่จะมารองรับ วงจรออกแบบจะถูกกำหนดขึ้น และระบบช่วยการออกแบบจะสังเคราะห์วงจรที่ได้จากรูปแบบที่เขียนขึ้นให้อยู่ในรูปของ วงจรที่ประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์ หรือวงจรในระดับเกท และการเชื่อมต่อระหว่างกันของอุปกรณ์เหล่านั้นหรือ ไม่ก็อยู่ในรูปของ Netlist ที่สามารถนำไปผลิตในอุปกรณ์อื่นได้

2.3.16.4 หลังจากการสังเคราะห์วงจรให้อยู่ในระดับเกทหรือ Netlist แล้วข้อมูลนี้จะถูกใช้สำหรับจำลองการทำงานในเรื่อง ความถูกต้องของฟังก์ชัน พร้อมกับนำข้อมูลที่เกี่ยวข้องกับเวลาเข้ามาประกอบการพิจารณาด้วย ซึ่งตามปกติแล้วอุปกรณ์ ทางอิเล็กทรอนิกส์ทุกชิ้นจะมีเวลาหน่วงของการแพร่กระจาย (Propagation Detry Time) เสมอ ถึงแม้ว่าจะเป็นเวลาที่น้อยมากในระดับนาโนวินาทีก็ตาม แต่ถ้าภายในวงจรหนึ่งประกอบด้วยเกทของฟังก์ชันต่างๆ จำนวน 10,000 เกท ขึ้นไป เวลาดังกล่าวนี้จะสะสมกันมากขึ้น จนอาจทำให้การทำงานของวงจรรวมทั้งหมดผิดพลาดไปหรือไม่สามารถทำงานในย่านความถี่สัญญาณนาฬิกาที่สูงได้

2.3.16.5 ผลิตเป็นวงจรจริง (Technology and device mapping) โดยนำข้อมูลที่ได้จากการสังเคราะห์มาผลิตซึ่งอาจจะอยู่ในรูปของแผงวงจรไฟฟ้า ที่ประกอบด้วยอุปกรณ์หลายๆ ชิ้นหรืออยู่ในรูปของวงจรรวม ASIC

2.3.16.6 ทำการตรวจสอบการทำงานและตัวแปรทางด้านเวลาทั้งหมด เพื่อความถูกต้องของวงจรเป็นครั้งสุดท้ายก่อนนำไปรวมเข้ากับอุปกรณ์อื่นๆ ให้เป็นระบบดิจิตอล เนื่องจากในขั้นตอนนี้ วงจรที่ออกแบบจะประกอบด้วยจุดต่อทางอินพุทและเอาท์พุท ซึ่งเป็นจุดต่อสำหรับการรับและส่งสัญญาณกับภายนอก

2.3.16.7 นำวงจรที่ออกแบบไว้ประกอบเข้ากับอุปกรณ์อื่นๆ ให้เป็นระบบที่สมบูรณ์ แล้วทำการทดสอบการทำงานทั้งระบบร่วมกับอุปกรณ์อื่นๆ อีกครั้งเพื่อควบคุมคุณภาพของผลิตภัณฑ์

บทที่ 3
การออกแบบ

จอมอนิเตอร์ที่จะใช้เครื่องตรวจเช็คจอเครื่องนี้เป็นเครื่องทดสอบจอที่นำ FPGA มามีส่วนช่วยในการออกแบบสัญญาณในการทดสอบจอมอนิเตอร์

โดยทั่วไปแล้วสัญญาณที่ใช้ทดสอบจอมอนิเตอร์จะเป็นสัญญาณที่ทดสอบการแสดงสีแต่ละสี โดยทำการส่งสัญญาณสีแต่ละสีมาทดสอบที่ละสี แล้วสังเกตผลว่าแต่ละสีทำงานเป็นปกติหรือไม่ ซึ่งถ้ามีความผิดปกติในการแสดงสีจะได้นำไปแก้ไขต่อไป

การออกแบบจะเน้นอยู่ที่มีอุปกรณ์น้อยที่สุด วงจรไม่ซับซ้อน อุปกรณ์ทุกตัวหาง่าย

เพียงเสียบเข้าคอนเน็กเตอร์ของจอมอนิเตอร์ จ่ายไฟเข้าเครื่อง แล้วกดสวิตซ์เท่านั้น ก็สามารถรู้ได้เลยว่าจอที่ท่านกำลังทดสอบอยู่นั้นดีหรือเสีย

วงจรที่สมบูรณ์แบบของเครื่องเช็คจอมอนิเตอร์ (Computer monitor checker) ปรากฏอยู่ในรูป โดยวงจรดังกล่าวประกอบด้วย 3 ส่วน คือ

1.วงจรผลิตความถี่ในการสร้างสัญญาณภาพและวงจรภาคซิงค์ วงจรส่วนนี้ทำหน้าที่เป็นหน่วยผลิตความถี่ ทำหน้าที่สร้างสัญญาณซิงค์เพื่อให้การสแกนของเส้นภาพเป็นไปอย่างถูกต้อง ได้จำนวนเส้นภาพและรายละเอียดภาพตามคาดหวัง วงจรในส่วนนี้มีไอซีทำหน้าที่หารความถี่ และจัดคาบเวลาให้เป็นตามมาตรฐาน VGA และอื่นๆ ซึ่งก็อยู่ในส่วนการโปรแกรมลง FPGA เช่นกัน

2.วงจรควบคุม ทำหน้าที่ควบคุมการส่งสัญญาณแม่สีและซิงค์ออกไปยังซ็อกเก็ต โดยมีสวิตช์ตรวจสอบการทำงาน การควบควบคุมความถี่ที่ส่งเข้ามา การควบคุมคาบเวลาในการทำงาน การเอาสัญญาณจากซ็อกเก็ตแบบดีคอนเน็คเตอร์ไปยังจอมอนิเตอร์ โดยมีFPGA เป็นตัวควบคุมการทำงานทั้งหมด

3. ภาคจ่ายไฟ จะมีอยู่ 2 ส่วน คือ ที่บอร์ด FPGA และส่วนแหล่งจ่ายไฟของวงจรภาคสร้างสัญญาณขึ้นบันได 9โวลต์โดยมีไอซีเรกูเลเตอร์3ขาทำหน้าที่เป็นวงจรรักษาระดับแรงดันไฟคงที่ 5 โวลต์เพื่อจ่ายให้กับวงจร

3.1 วงจรผลิตความถี่และสัญญาณซิงค์

วงจรผลิตความถี่ของระบบใช้วงจรคริสตัลออสซิลเลเตอร์ผลิตความถี่ 25.125 เมกะเฮิรตซ์ ป้อนให้กับ FPGA แล้วใช้การเขียนโปรแกรมเพื่อนำความถี่ที่รับเข้ามานี้มาสร้างลูกคลื่นสัญญาณ RGB และสัญญาณซิงค์ทั้งสองสัญญาณ

โดยมี IC สร้างความถี่มาตรฐานส่งเข้ามาที่ตัว FPGA

โดยที่ความละเอียดของหน้าจอที่จะใช้คือ 640x480 มีรายละเอียดคือ

ต้องการสัญญาณนาฬิกาภายในจอภาพ(Internal Monitor Clock) สำหรับสร้างสัญญาณวิดีโอ 25.175 เมกะเฮิรตซ์

#จำนวนจุดมากสุดต่อเส้น(max dots per line) = Dot clock frequency/Horizontal Sync

= 25.175 MHz / 31.5kHz

= 800 จุด

เราจะใช้แค่ 640 จุด จาก 800 จุด เนื่องจากเราต้องการพื้นที่ขอบซ้ายและขวาและเวลาสำหรับ Horizontal retrace

#จำนวนเส้นมากสุดต่อหน้าจอ = Horizontal Sync/Vertical Sync

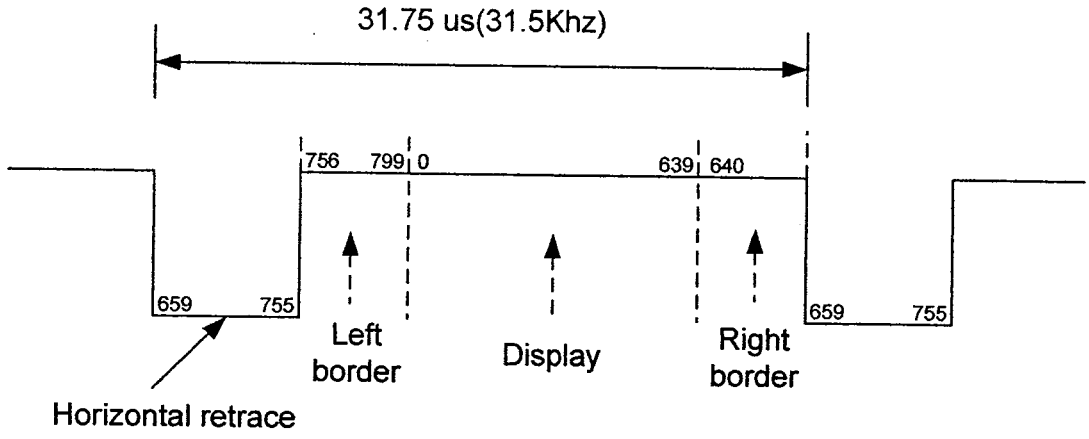
= 31.5kHz / 60hz = 525 เส้น

เราจะใช้แค่ 480 เส้น เพื่อการสร้างพื้นที่ขอบบนล่างและช่วงเวลาสำหรับ Vertical retrace

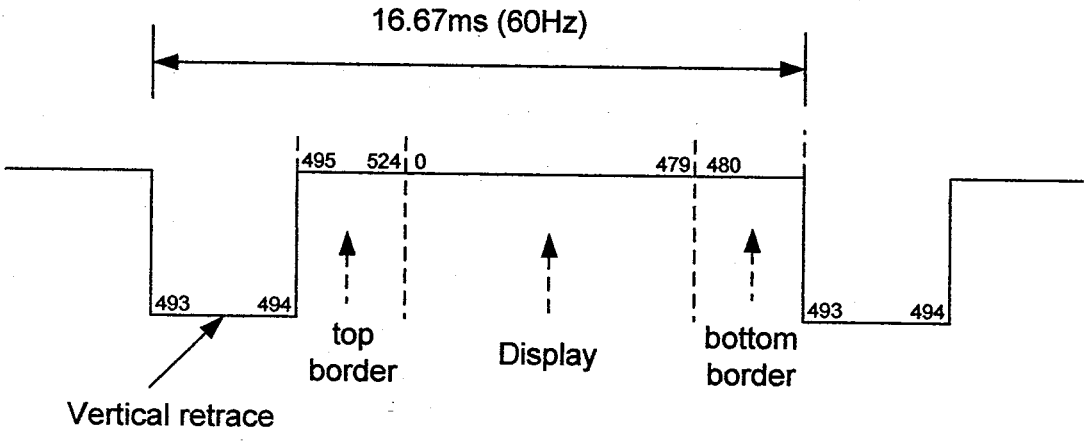
คาบเวลาของการรีเทรซ

เรื่องของมอนิเตอร์มีรายละเอียดเกี่ยวกับคาบเวลาของสัญญาณซิงค์เพิ่มเติมเข้ามาอีก เป็นเรื่องของคาบเวลาที่แต่ละระบบมีความต่างกันออกไป 5-20 ไมโครวินาที ทั้งนี้เพราะเวลาของการสะบัดเส้นภาพกลับที่ไม่เท่ากัน

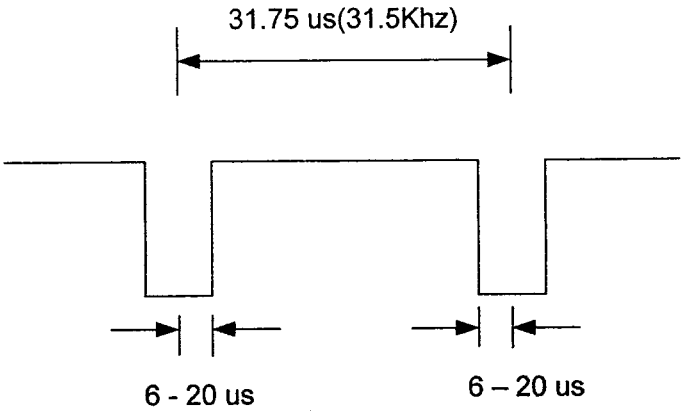
การบังคับลำอิเล็กตรอนให้สะบัดกลับเรียกเวลาในจังหวะนี้ว่าการรีเทรซ เมื่อเวลาที่จะให้ภาพปรากฏมีคาบเวลาต่างกันออกไป เวลาของการแบลงกิ้งหรือการทำให้ขอบจอมืดแค่ไหนเรียกว่าการรีเทรซ กำหนดเวลาของการเกิดแสงจากตรงไหนของขอบจอเรียกว่าเวลาของการเทรซหรือ Grater time period เมื่อเวลาในส่วนนี้ต่างกัน



รูปที่ 3.1 Horizontal Timing



รูปที่ 3.2 Vertical Timing



รูปที่ 3.3 สัญญาณ สีแดง สีเขียว สีน้ำเงิน กรณีแสดงสีขาว

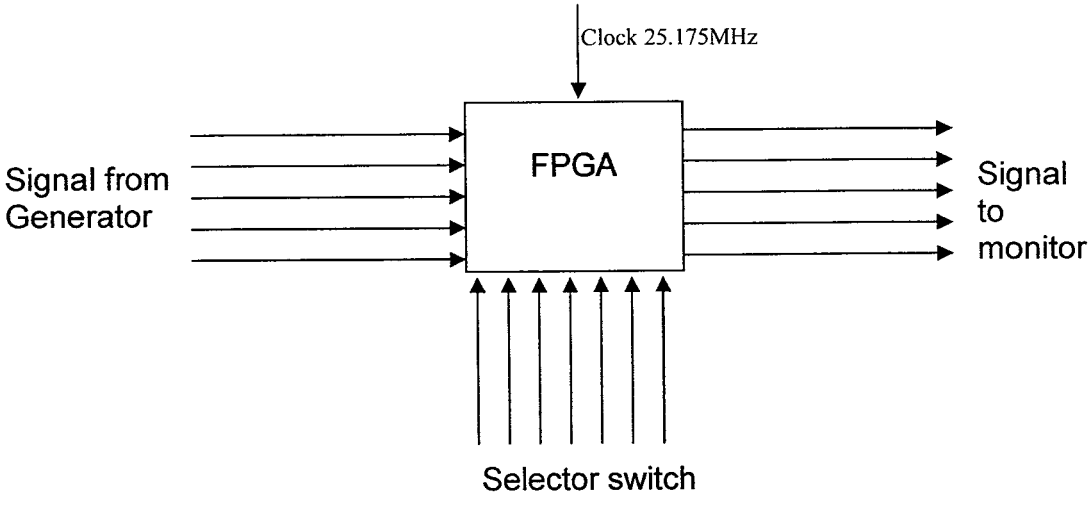
ส่วนของความถี่สัญญาณสีและสัญญาณซิงค์ใช้ IC MSP430F1111 ในการสร้างความถี่เพื่อป้อนให้ FPGA ต่อไปโดยจะสร้างความถี่ของสีขึ้นมาทั้งสามสี(RGB)โดยมีความถี่ประมาณ 31.5 KHz ,ความถี่ Horizontal และ Vertical Sync ซึ่งมีความถี่ประมาณ 31.5KHz และ 60Hz ตามลำดับ และมีสวิตซ์ ที่ขา9 ของ IC ที่จะทำการทริกให้ IC เปลี่ยนรูปคลื่นสัญญาณ RGB โดยสามารถดูรายละเอียดวงจรได้ที่ภาคผนวก

3.2 การออกแบบวงจรเอาต์พุต

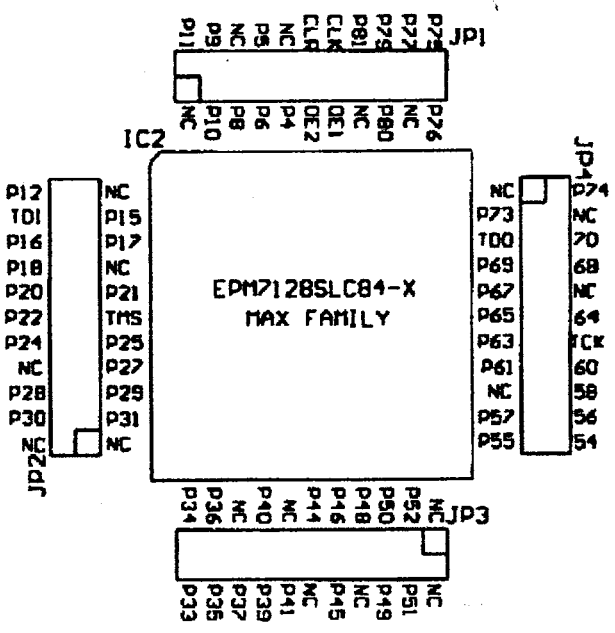
การออกแบบวงจรทางด้านเอาต์พุตจะให้สามารถรองรับระบบมาตรฐานของมอนิเตอร์ ซึ่งจะรองรับสัญญาณ RGB ที่ระดับประมาณ 0-0.7 โวลต์ โดยต่อจาก FPGA ทางด้านเอาต์พุตที่ขา 30 เป็นสัญญาณเวอร์ติคอล ขา 28 เป็นสัญญาณฮอร์ริซอนตอล ส่วนขา 31 เป็นสัญญาณ R (RED) , ขา 29 เป็นสัญญาณ G (GREEN) และขา 27 เป็นสัญญาณ B (BLUE)

จะใช้ FPGA เบอร์ EPM7128SLC84-15 ซึ่งเป็นแบบ CPLD(สามารถเก็บโปรแกรมได้โดยไม่ต้องใช้แหล่งจ่ายไฟ) เป็นตัวควบคุมการแสดงผลที่จอมอนิเตอร์ โดยที่ FPGA จะรับค่ามาจากสวิตช์เลือกฟังก์ชันในการส่งสัญญาณในการส่งสัญญาณออกไปที่จอมอนิเตอร์โดยที่ FPGA จะมีความถี่ที่ป้อนให้ 25.175MHz (39.72nSec) นำความถี่นี้มาใช้ในการจัดสัญญาณในการส่งความถี่ออกไป

FPGAจะรับค่าจากสวิตซ์มาก่อนแล้วจึงนำClockที่มาจากส่วนผลิตความถี่ แล้วทำการเปรียบเทียบกับสัญญาณ 25.175MHz ที่FPGAแล้วได้เป็นรูปคลื่นสัญญาณทั้ง 5 สัญญาณ(RGB และ Sync)

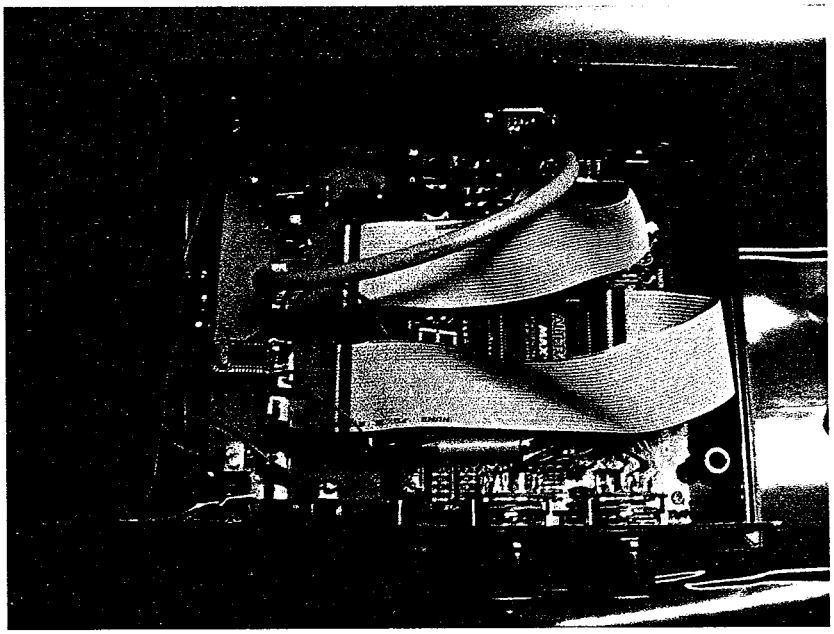


รูปที่ 3.4 แสดงการติดต่อของ FPGA กับสัญญาณต่างๆ

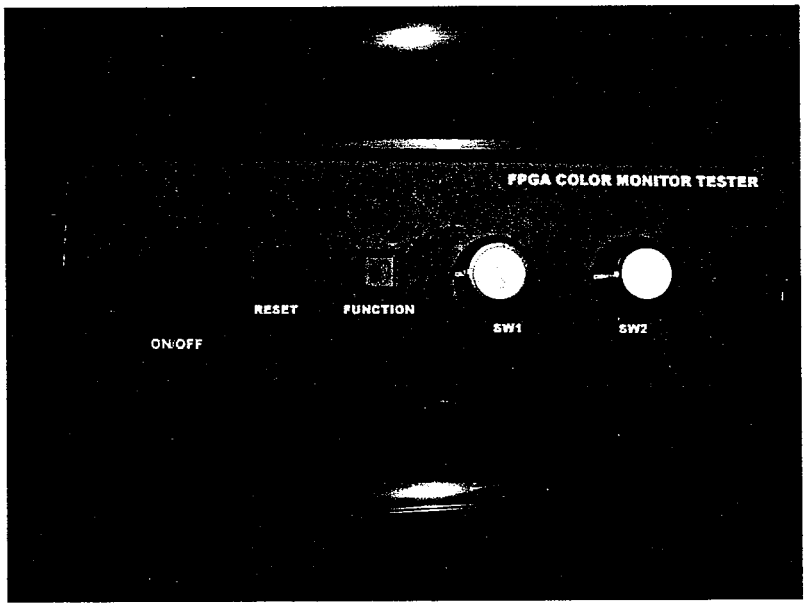


รูปที่ 3.5 แสดงการต่อพอร์ทจาก FPGA ไปยัง DB15 เพื่อแสดงผลที่จอมอนิเตอร์

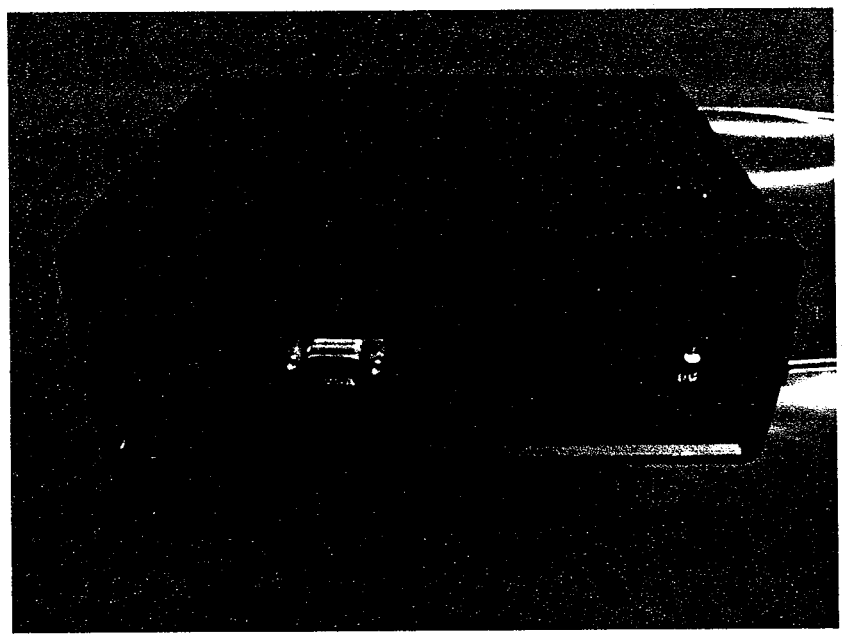
3.3 ลักษณะชิ้นงาน



รูปที่ 3.6 แสดงด้านบนของชิ้นงาน



รูปที่ 3.7 แสดงด้านหน้าของชิ้นงาน



รูปที่ 3.8 แสดงด้านหลังของชิ้นงาน

บทที่ 4

ผลการทดลอง

บทนี้กล่าวถึงการทดลองต่างๆและผลการทดลองที่ได้แล้วในโครงงานนี้ โดยทำการวัดสัญญาณที่จุดต่างๆ แล้วบันทึกค่าไว้เปรียบเทียบกับค่าที่ได้ทางทฤษฎี

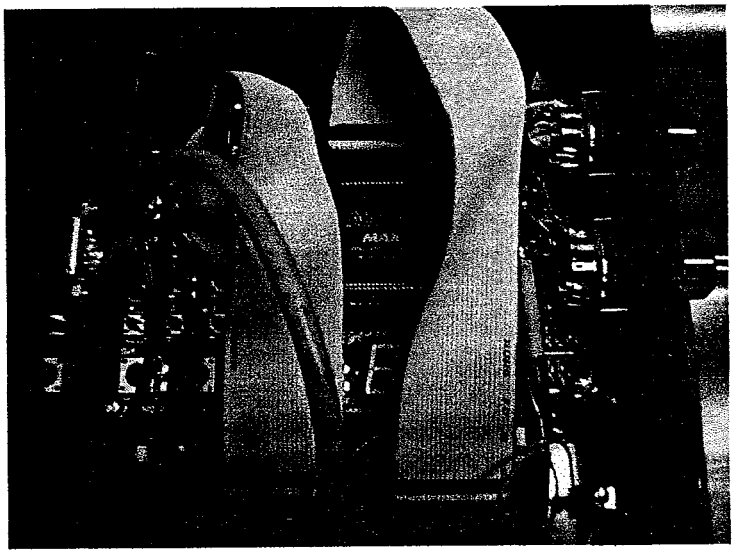
4.1 วัตถุประสงค์การทดลอง

- เพื่อเปรียบเทียบค่าที่ได้จากการทดลองว่าเป็นไปตามทฤษฎีหรือไม่
- เพื่อเป็นแนวทางในการวิเคราะห์หาจุดบกพร่องของโครงงาน

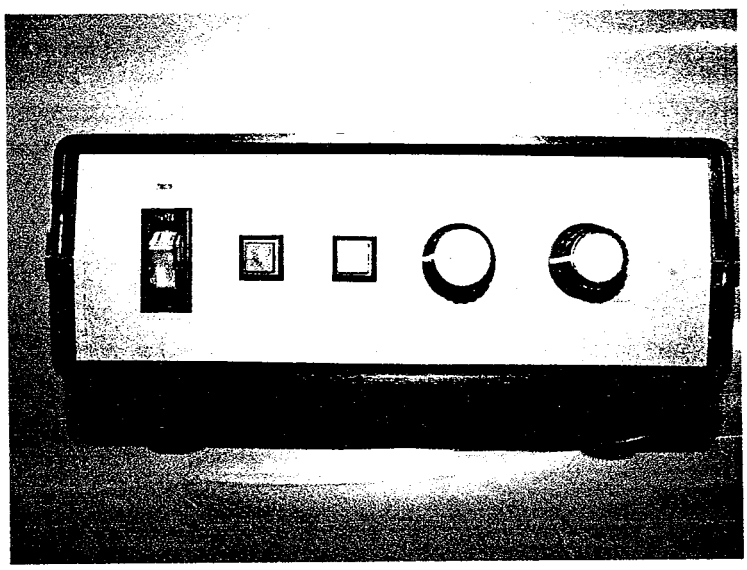
4.2 อุปกรณ์การทดลอง

- บอร์ด FPGA PLD A01(EPM7128SLC84-15)
- เครื่องคอมพิวเตอร์
- จอมอนิเตอร์
- Software MAX PLUS II

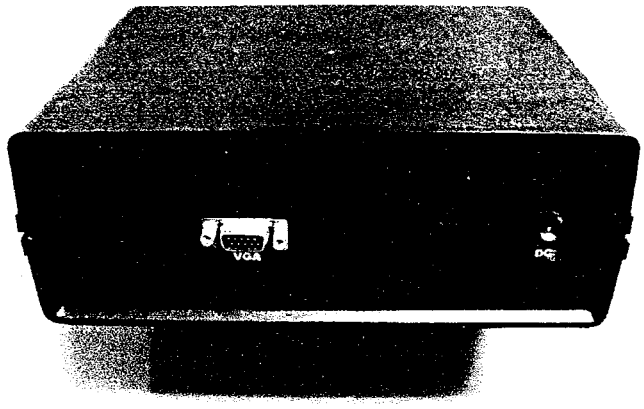
4.3 การทดลองใช้งาน



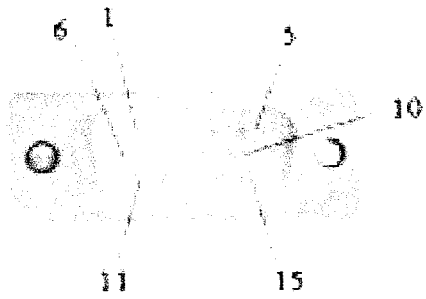
รูปที่ 4.1 แสดงบอร์ด FPGA PLD A01(EPM7128SLC84-15)



รูปที่ 4.2 แสดงด้านหน้าของชิ้นงาน สวิทช์เลือกฟังก์ชัน

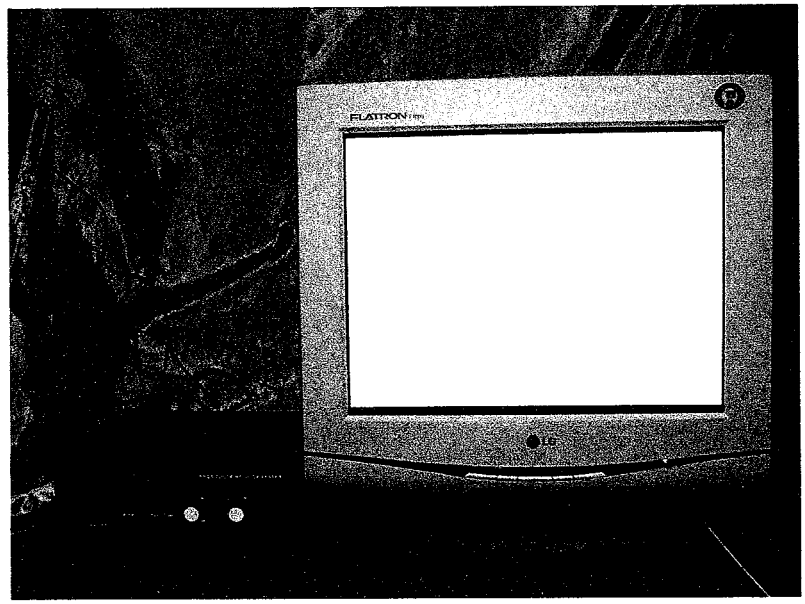


รูปที่ 4.3 แสดงด้านหลังของชิ้นงาน ช่องต่อใช้งาน และช่องต่อแหล่งจ่ายไฟ



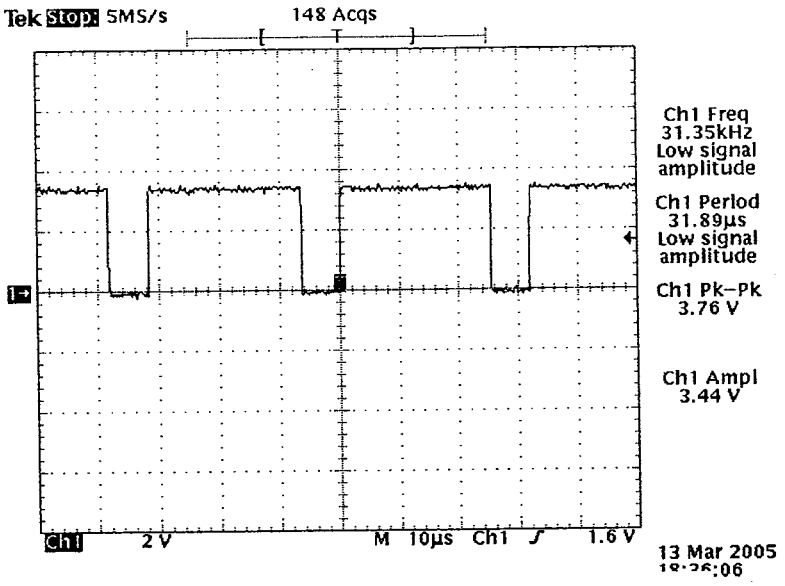
รูปที่ 4.4 แสดงจุดวัดสัญญาณ

4.3.1เลือกฟังก์ชันสีขาว



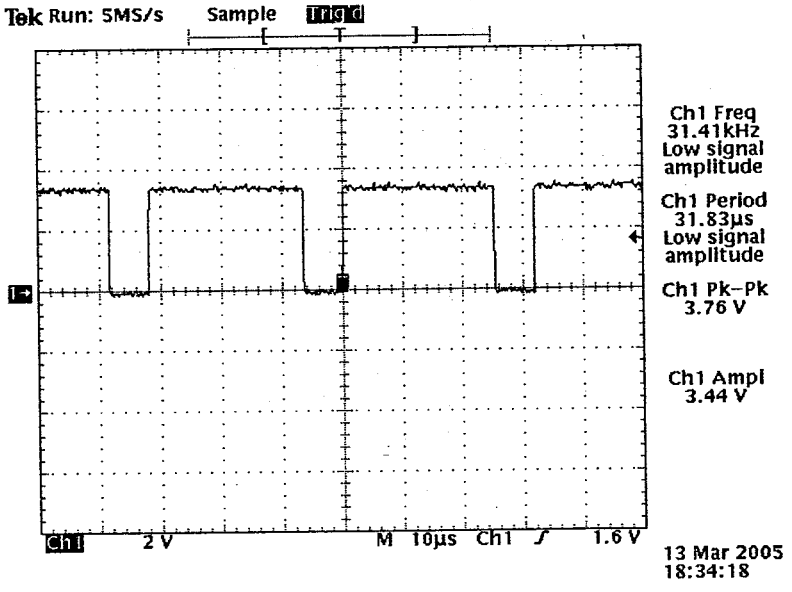
รปที่ 4.5 แสดงผลการทำงานฟังก์ชันสีขาว

4.3.1.1 จุดวัดสัญญาณเอาต์พุตที่ขา 1 (RED) ของ DB15



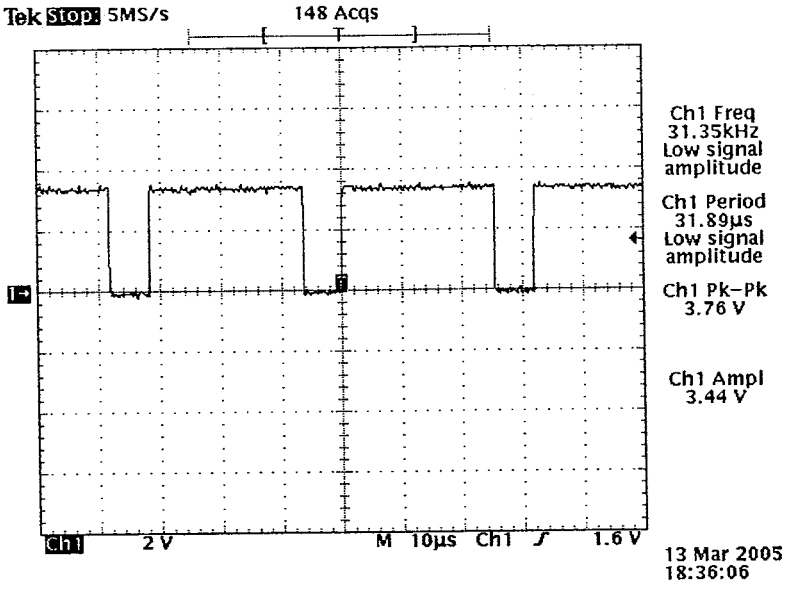
รูปที่ 4.6 รูปสัญญาณสีแดง

4.3.1.2 จุดวัดสัญญาณเอาต์พุตที่ขา 2 (GREEN) ของ DB15



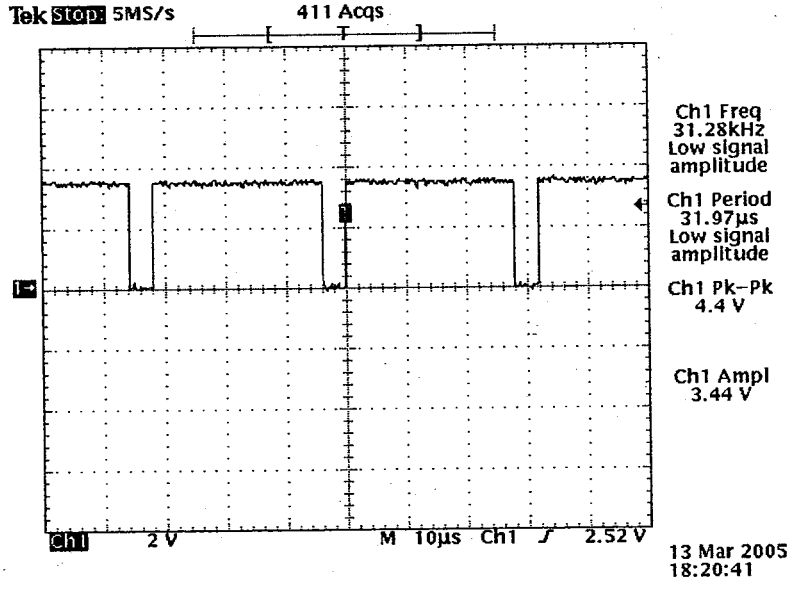
รูปที่ 4.7 รูปสัญญาณสีเขียว

4.3.1.3 จุดวัดสัญญาณเอาต์พุตที่ขา 3 (BLUE)ของ DB15



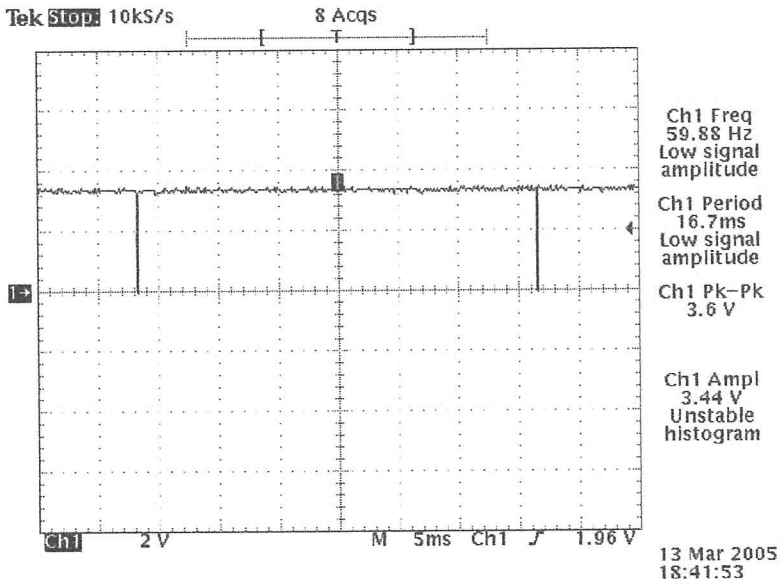
รูปที่ 4.8 รูปสัญญาณสีน้ำเงิน

4.3.1.4 จุดวัดสัญญาณเอาต์พุตที่ขา HORของ DB15



รูปที่ 4.9 รูปสัญญาณ Horizontal Sync

4.3.1.5 จุดวัดสัญญาณเอาต์พุตที่ขา VER ของ DB15

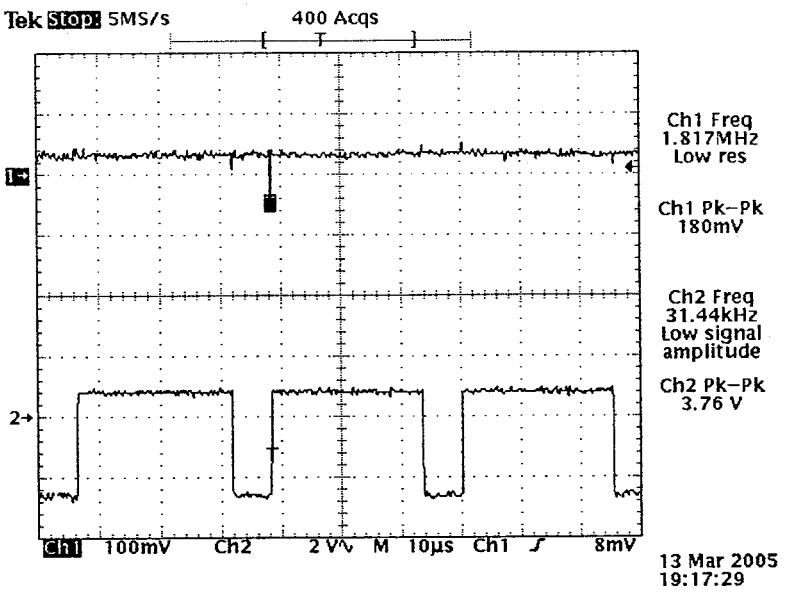


รูปที่ 4.10 รูปสัญญาณ Vertical sync

4.3.2 เลือกฟังก์ชันสีแดง

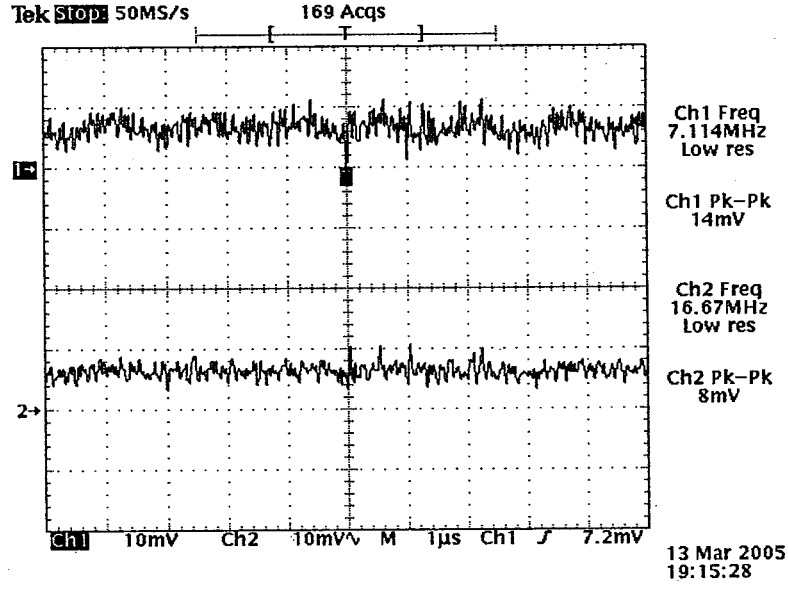
รปที่ 4.11 แสดงผลการทำงานฟังก์ชันสีแดง

4.3.2.1 จุดวัดสัญญาณเอาต์พุตที่ขา 2 (GREEN) และ 1 (RED) ของ DB15



รูปที่ 4.12 รูปสัญญาณสีเขียวเทียบสีแดง

4.3.2.2จุดวัดสัญญาณเอาต์พุตที่ขา 2 (GREEN) และ 3 (BLUE) ของ DB15

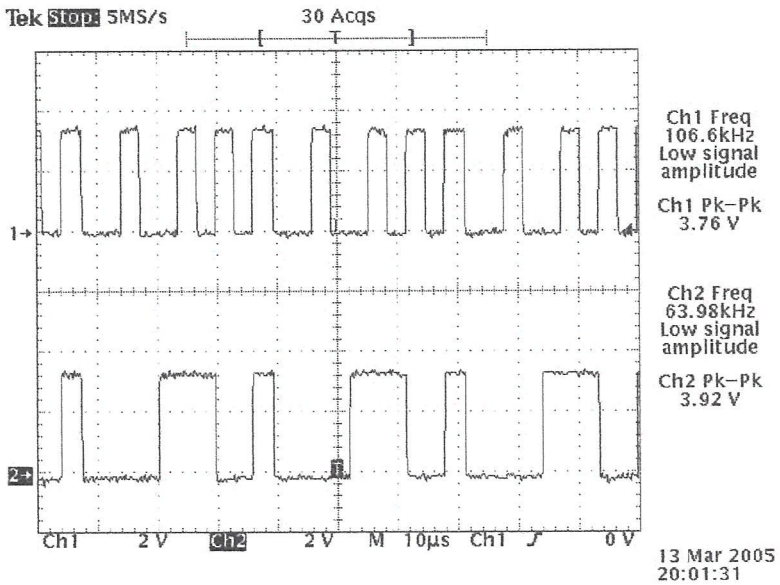


รูปที่ 4.13 รูปสัญญาณสีเขียวเทียบสีน้ำเงิน

4.3.3 เลือกฟังก์ชันแถบสี

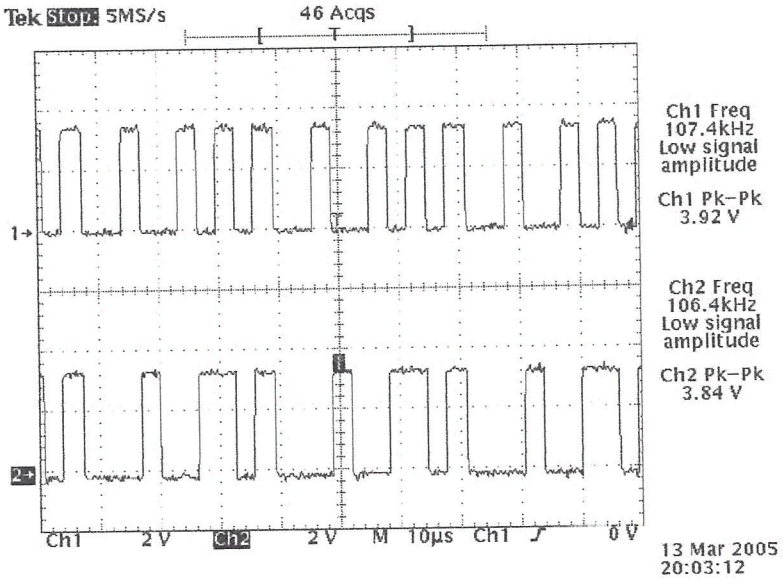
รปที่ 4.14 แสดงผลการทำงานฟังก์ชันแถบสี

4.3.3.1จุดวัดสัญญาณเอาต์พุตที่ขา 1 (RED) และ 2 (GREEN) ของ DB15



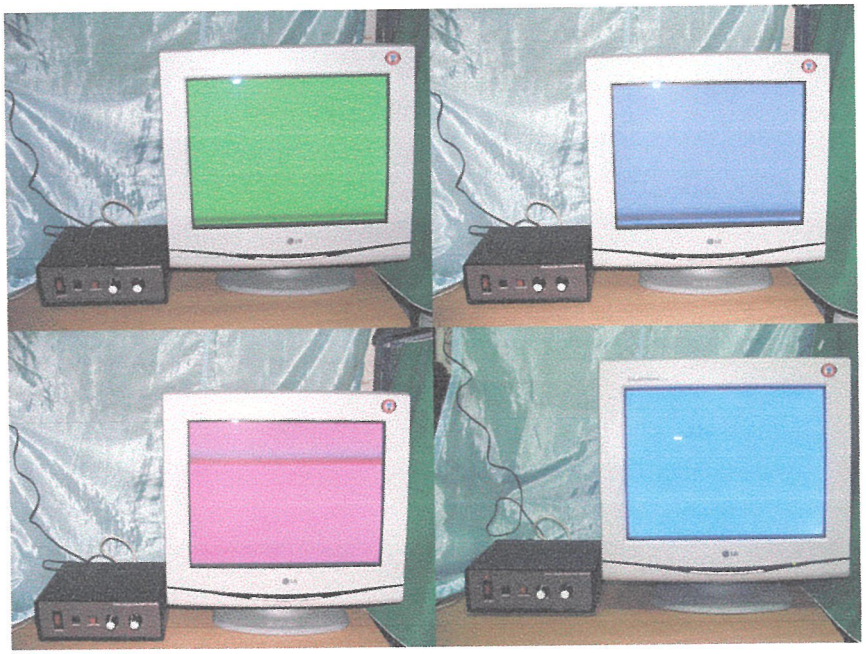
รูปที่ 4.15 รูปสัญญาณสีแดงเทียบสีเขียว

4.3.3.1 จุดวัดสัญญาณเอาต์พุตที่ขา 1 (RED) และ 3 (BLUE) ของ DB15



รูปที่ 4.16 รูปสัญญาณสีแดงเทียบสีน้ำเงิน

4.3.4 ฟังก์ชันอื่นๆ



รปที่ 4.17 แสดงผลการทำงานฟังก์ชันอื่นๆ

4.4 วิเคราะห์ผลการทดลอง

จากการทดลองที่ 4.3.1 เป็นการแสดงหน้าจอมอนิเตอร์จะเป็นสีขาว โดยที่เมื่อวัดสัญญาณ R (สีแดง) เทียบกับสัญญาณ G (สีเขียว) และสัญญาณ B (สีน้ำเงิน) โดยที่ป้อนสัญญาณ R, G, B ที่มีรูปคลื่นเหมือนกันและช่วงเวลาเดียวกันทั้งสามเข้ายังจอมอนิเตอร์จะได้การแสดงผลที่หน้าจอมอนิเตอร์ซึ่งภาพที่ได้จะเป็นสีขาว แรงดันที่วัดได้R, G, B มีค่า 3.76 V ส่วนสัญญาณฮอริซอนตอลที่วัดได้จะมีช่วงความถี่ที่ได้คือ 31.28 kHz ซึ่งมีค่าใกล้เคียงกับทฤษฎีที่จอมอนิเตอร์ต้องการคือ 31.5 kHz และสัญญาณเวอร์ติคอลที่วัดได้จะมีช่วงความถี่ที่ได้คือ 59.88 Hz ซึ่งมีค่าใกล้กับทฤษฎีที่จอมอนิเตอร์ต้องการคือ 60 Hz ซึ่งเป็นขนาดภาพ 640x480 pixels

จากการทดลองที่ 4.3.2 เป็นการแสดงหน้าจอมอนิเตอร์จะเป็นสีแดง โดยที่เมื่อวัดสัญญาณ G (สีเขียว) เทียบกับสัญญาณ R (สีแดง) และสัญญาณ B (สีน้ำเงิน) โดยที่การกำหนดการเกิดสัญญาณที่สัญญาณ R (สีแดง) ค่าแรงดันที่วัดได้มีค่าเป็น 3.76 v และสัญญาณ G (สีเขียว), B(สีน้ำเงิน) มีค่าเป็น 0 v ส่งผลให้ภาพที่แสดงหน้าจอมอนิเตอร์เป็นสีแดง

จากการทดลองที่ 4.3.3 เป็นการแสดงหน้าจอมอนิเตอร์เป็นแถบสี โดยที่เมื่อวัดสัญญาณสัญญาณ R (สีแดง) เทียบกับ G (สีเขียว) และสัญญาณ B (สีน้ำเงิน) โดยที่การกำหนดการเกิดสัญญาณที่สัญญาณ R (สีแดง), G (สีเขียว) และ B (สีน้ำเงิน) เพื่อให้เกิดภาพแถบสีเรียงกันจะได้สีที่เรียงกันคือ สีแดง, สีเขียว, สีน้ำเงิน, สีม่วงแดง(สีแดง + สีน้ำเงิน), สีฟ้า(สีน้ำเงิน + สีเขียว), สีเหลือง (สีแดง + สีเขียว), สีดำ(ค่า R,G,B มีค่าเป็น 0 v) และสีขาว(ค่า R,G,B มีค่าสัญญาณเหมือนกัน)

บทที่ 5
สรุปผลการทดลองและปัญหาที่พบ

5.1 สรุปผลการทดลอง

จากการทำงานของวงจรจะมีการใช้ความถี่ 25.175 MHz มาเป็นตัวกำหนดการทำงานของสัญญาณสีแดง สีเขียว สีน้ำเงิน สัญญาณเวอร์ติคอลซิงค์ และฮอริซอนตอลซิงค์ซิงค์ โดยใช้การเขียนโปรแกรมที่ FPGA กำหนดช่วงเวลาในการทำงาน การตรวจสอบสวิตช์ที่กำหนดฟังก์ชันการทำงานไว้ แล้วทำการส่งสัญญาณออกไปเพื่อไปยังจอมอนิเตอร์

5.2 ปัญหาที่พบ

5.2.1 ปัญหาด้านการเกิดดีเลย์ของสัญญาณที่ออกมามีความถี่และคาบเวลาไม่ตรงกับการใช้โปรแกรมจำลองการทำงาน Max Plus 2

5.2.2 ปัญหาด้านการออกแบบการทำงานของสวิตช์กับสัญญาณที่ออกไปได้สัญญาณที่ไม่สมบูรณ์

5.2.3 ปัญหาด้านความถี่คริสตัลที่ส่งความถี่ไม่ได้ตามต้องการ

5.2.4 ปัญหาเรื่องอุปกรณ์การทดลอง จำเป็นต้องใช้ออสซิลโลสโคปที่มีการอ่านความถี่และคาบเวลาได้อย่างชัดเจน มิฉะนั้นความถี่ที่วัดได้จะไม่เที่ยงตรง และบางครั้งก็วัดสัญญาณได้ไม่สมบูรณ์

5.3 แนวทางในการแก้ปัญหา

5.3.1 ใช้อุปกรณ์ ไอซีมาช่วยในการสร้างความถี่และคาบเวลาในช่วงที่กำหนด

5.3.2 นำโครงสร้างไอซีมาต่อวิเคราะห์โดยใช้ Software MAX PLUS II วิเคราะห์ก่อนนำไปต่อใช้งานจริง

5.3.3 พยายามคำนวณการทำงานที่โปรแกรมให้ใกล้เคียงกับที่ตัวคริสตัลส่งมาให้จริง

หนังสืออ้างอิง

[1].ชำนาญ ปัญญาใส และ วัชรากร หนูทอง "ภาษา VHDL สำหรับการออกแบบวงจรดิจิตอล", ซีเอ็ดยูเคชั่น, กรุงเทพฯ พ.ศ. 2547

[2].บริษัท Astron logic, " เปิดโลก FPGA กับ Wizard PLD-A01".

[3] S. Sjoholm and L.Lindh, "VHDL for Designers", Prentice Hall, 1997.

[4] S.Brown and Z.Vranesic, "Fundamentals of digital logic with VHDL", McGraw-HILL, 2000.

[5] K.S. Lin "Texas Instruments Digital Signal Processing", Prenctice Hall, pp.(F)2-(F)9, 1998.

[6] C.E. Sporck, "National Semiconductor F100K ECL Logic Databook", pp.(3)199-(3)205, 1990.

[7] www.astronlogic.com

[8] www.datasheetarchive.com

ภาคผนวก

ภาคผนวก ก.

บอร์ดทดลอง PLD - A01

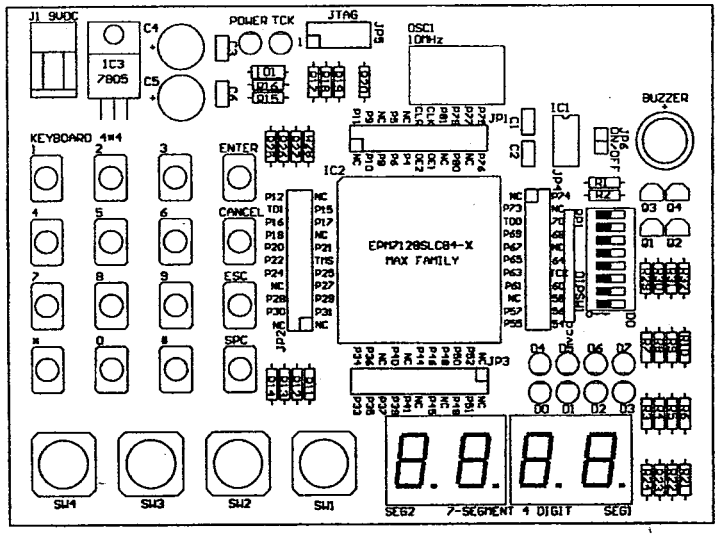
บอร์ดทดลอง

PLD – A01

PLD-A01/1 เป็นบอร์ดทดลองที่ถูกออกแบบมาเพื่ออำนวยความสะดวกในการเรียนรู้และใช้งาน FPGA ของบริษัท ALTERA ในตระกูล MAX7000S อนุกรม EPM7128SLC84 โดยใช้งานร่วมกับโปรแกรม MAX+PLUS II ในการออกแบบและพัฒนาระบบทางดิจิตอลระดับสูง

โครงสร้างและองค์ประกอบของบอร์ด PLD - A01

ส่วนประกอบต่างๆของบอร์ด PLD - A01 เป็นดังรูปที่ A1



รูปที่ A1 บอร์ด PLD - A01/1

อุปกรณ์ต่าง ๆบนบอร์ด PLD - A01 มีดังต่อไปนี้

- วงจรรวมตระกูล MAX7000S ในอนุกรม EPM7128SLC84
- JTAG CONNECTOR
- พอร์ตขยายช่องสัญญาณ
- สวิตช์ กดติด - ปล่อยดับ 4 ตัว
- สวิตช์เมตริกซ์ขนาด 4 x 4
- ดิพสวิตช์ 8 หลัก 1 ชุด
- ไดโอดเปล่งแสง 8 ดวง

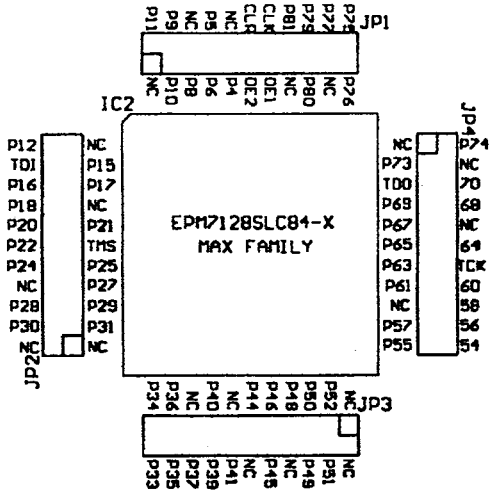
- 7-Segment 4 หลักชนิด Common Anode แบบ Multiplex
- BUZZER 1 ชุด
- OSCILLATOR ความถี่ 10.000 MHz
- DC INPUT แรงดัน 7-10 Volt

วงจรรวมตระกูล MAX7000S ในอนุกรม EPM7128SLC84

เป็นวงจรรวมที่มีความจุปานกลาง ประมาน 2,500 เกต โดยมีโครงสร้างภายในเป็นแบบ EEPROM BASE FPGA นอกจากนี้สามารถเก็บข้อมูลที่โปรแกรมลงไปได้โดยไม่จำเป็นต้องมีไฟเลี้ยง (Non-Volatile Configuration) ซึ่งจะใช้ AND-OR Plane ในการทำลอจิกฟังก์ชัน และมักมีการจัดสถาปัตยกรรมในรูปแบบอะเรย์ (Array Structure) ส่วนการโปรแกรมสามารถทำซ้ำได้ประมาน 10,000 ครั้ง ภายใน EPM7128SLC84 จะประกอบด้วย Macrocell จำนวน 128 ตัว ในแต่ละ Macrocell จะมี Programmable-AND/Fixed-OR Array, Configurable Register with Independently Programmable clock, Clock Enable, Clear และ Preset Functions

พอร์ตขยายช่องสัญญาณ

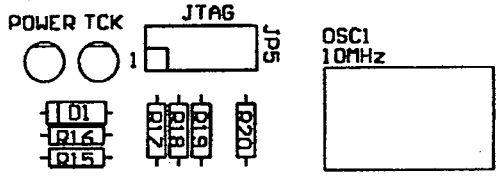
เป็น Header คู่ตัวเมียสำหรับขยายการต่อช่องสัญญาณตำแหน่งขาของชีพ ไอซี EPM7128SLC84 กับ Hole จะมีความสัมพันธ์ดังรูปที่ A2



รูปที่ A2 การจัดวางตำแหน่งของชิพไอซี กับ Header

JTAG Connector

ใช้สำหรับต่อสาย Byte Blaster เพื่อ Download ข้อมูลของวงจรลอจิกจาก Computer ลงในชิพไอซี ซึ่งจะมีตำแหน่งขาและการจัดวางดังรูปที่ A3



รูปที่ A3 ตำแหน่งขาและการจัดวาง JTAG Connection

สวิตช์กดติด -ปล่อยดับ 4 ตัว

สวิตช์ SW1-SW4 เป็นสวิตช์แบบกดติด - ปล่อยดับ และมีตัวต้านทาน PULL-UP 10k ต่ออยู่ เมื่อกดสวิตช์จะได้สัญญาณลอจิกศูนย์ โดยที่

SW1 ต่อกับขา 49 ของ EPM7128SLC84

SW2 ต่อกับขา 50 ของ EPM7128SLC84

SW3 ต่อกับขา 51 ของ EPM7128SLC84

SW4 ต่อกับขา 52 ของ EPM7128SLC84

สวิตช์เมตริกช์ขนาด 4x4

เป็นสวิตช์เมตริกช์ ขนาด 16 ปุ่ม ประกอบด้วย 4 Row กับ 4 Column ในหลักของ Column จะมีตัวต้านทาน PULL UP 10k ต่ออยู่และมีลักษณะการจัดวางวงจรดังรูปที่ A4 โดยที่

Column ที่ 1 มีตัวต้านทาน pull up 10 k ต่อกับขา 28 ของ EPM7128SLC84

Column ที่ 2 มีตัวต้านทาน pull up 10 k ต่อกับขา 29 ของ EPM7128SLC84

Column ที่ 3 มีตัวต้านทาน pull up 10 k ต่อกับขา 30 ของ EPM7128SLC84

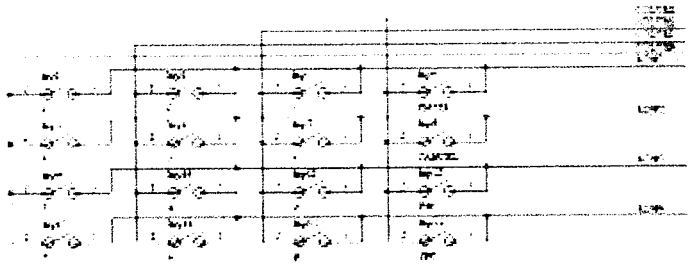
Column ที่ 4 มีตัวต้านทาน pull up 10 k ต่อกับขา 31 ของ EPM7128SLC84

Row ที่ 1 ต่อกับขา 22 ของ EPM7128SLC84

Row ที่ 2 ต่อกับขา 24 ของ EPM7128SLC84

Row ที่ 3 ต่อกับขา 25 ของ EPM7128SLC84

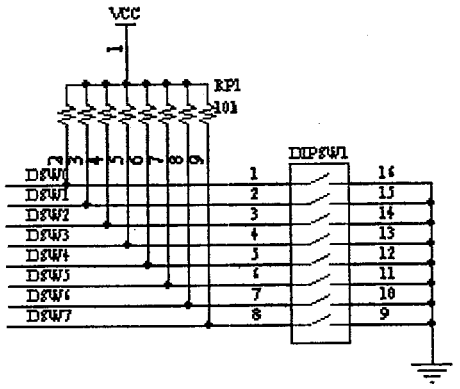
Row ที่ 4 ต่อกับขา 27 ของ EPM7128SLC84



*รูปที่ A4 วงจรสวิตช์เมตริกซ์ขนาด 4*4*

ดิพสวิตช์ 8 หลัก

เป็นดิพสวิตช์ขนาด 8 ช่อง (Octal Dip Switches) ต่ออยู่กับตัว ต้านทาน PULL UP 10k เมื่อเลื่อนสวิตช์ไปที่ตำแหน่ง ON จะได้สัญญาณลอจิกศูนย์ และเมื่อเลื่อนสวิตช์กลับมาที่ตำแหน่ง OFF จะได้สัญญาณลอจิกหนึ่งมีการต่อวงจรดังรูปที่ A5



รูปที่ A5 การต่อวงจรของดิพสวิตช์ 8 ช่อง

โดยที่

DSW0 ต่อกับขา 54 ของ EPM7128SLC84

DSW1 ต่อกับขา 55 ของ EPM7128SLC84

DSW2 ต่อกับขา 56 ของ EPM7128SLC84

DSW3 ต่อกับขา 57 ของ EPM7128SLC84

DSW4 ต่อกับขา 58 ของ EPM7128SLC84

DSW5 ต่อกับขา 60 ของ EPM7128SLC84

DSW6 ต่อกับขา 61 ของ EPM7128SLC84

DSW7 ต่อกับขา 63 ของ EPM7128SLC84

ไดโอดเปล่งแสง 8 ตัว

ในบอร์ดทดลองชุดนี้มีไดโอดเปล่งแสง 8 ดวง โดยแต่ละดวงจะมีตัวต้านทาน 1k เป็นตัวจำกัดกระแส เมื่อให้ลอจิกหนึ่งจะทำให้ LED สว่าง และเมื่อให้ลอจิกศูนย์จะทำให้ LED ดับ ลักษณะการต่อวงจรเป็นดังรูปที่ A6 โดยที่

LED0 ต่อกับขา 21 ของ EPM7128SLC84

LED1 ต่อกับขา 20 ของ EPM7128SLC84

LED2 ต่อกับขา 18 ของ EPM7128SLC84

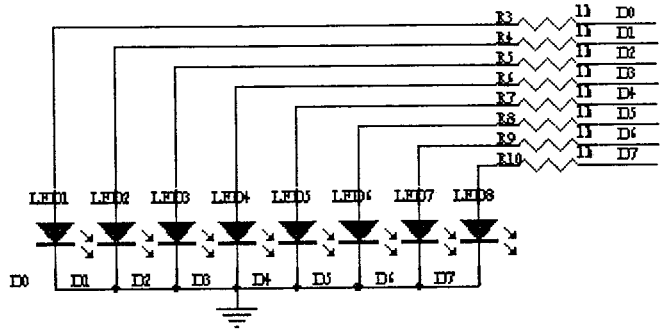
LED3 ต่อกับขา 17 ของ EPM7128SLC84

LED4 ต่อกับขา 16 ของ EPM7128SLC84

LED5 ต่อกับขา 15 ของ EPM7128SLC84

LED6 ต่อกับขา 12 ของ EPM7128SLC84

LED7 ต่อกับขา 11 ของ EPM7128SLC84



รูปที่ A6 การต่อวงจรสำหรับ LED 8 ดวง

7-Segment 4 หลักชนิด Common Anode แบบ Multiplex

SEG1 และ SEG2 เป็น LED 7-Segment ตัวละ 2 หลัก แบบคอมมอนแอโนด แต่ละหลักจะต่อเป็นแบบ Multiplexor สำหรับขาคอมมอนจะมีตัวต้านทาน 330 โอห์มต่ออนุกรมอยู่เพื่อจำกัดกระแสให้กับ LED หากต้องการให้ Segment a ของหลักที่หนึ่งสว่าง จะต้องให้ a เป็นลอจิกศูนย์ และเนื่องจากขาคอมมอนต่อกับทรานซิสเตอร์ชนิด PNP เพื่อที่จะทำให้ทรานซิสเตอร์นำกระแสจะต้องให้ c1 เป็นลอจิกศูนย์ จะทำให้ Segment A หลักที่หนึ่งสว่างขึ้นและจะมีตำแหน่งขาดังนี้

Segment a ต่อกับขา 33 ของ EPM7128SLC84

Segment b ต่อกับขา 34 ของ EPM7128SLC84

Segment c ต่อกับขา 35 ของ EPM7128SLC84

Segment d ต่อกับขา 36 ของ EPM7128SLC84

Segment e ต่อกับขา 37 ของ EPM7128SLC84

Segment f ต่อกับขา 39 ของ EPM7128SLC84

Segment g ต่อกับขา 40 ของ EPM7128SLC84

Segment dot ต่อกับขา 41 ของ EPM7128SLC84

C1 ต่อกับขา 44 ของ EPM7128SLC84

C2 ต่อกับขา 45 ของ EPM7128SLC84

C3 ต่อกับขา 46 ของ EPM7128SLC84

C4 ต่อกับขา 48 ของ EPM7128SLC84

BUZZER

ใช้ไอซี NE555 เป็นตัวกำเนิดความถี่เสียงให้แก่ BUZZER โดยใช้จุดต่อ SP ซึ่งต่ออยู่กับขา 10 ของ EPM7128SLC84 เมื่อ SP ได้รับลอจิกหนึ่งจะทำให้ BUZZER มีเสียงดังขึ้นมา และเมื่อให้ SP ได้รับลอจิกศูนย์จะทำให้ BUZZER ไม่ทำงาน ส่วน JP6 ไว้ สำหรับตัด BUZZER ออกจาก NE555 ซึ่งจะทำให้ไม่มีสัญญาณส่งให้ BUZZER เลย หากไม่ต้องการให้ BUZZER มีเสียงดังก็ให้ดึง JP6 ออก

- SP ต่อกับขา 10 ของ EPM7128SLC84

OSCILLATOR

บอร์ดทดลองชุดนี้ใช้ OSILLATOR แบบโมดูลออสซิเลเตอร์สำหรับผลิตความถี่ 10.000 MHz ป้อนให้กับขา 83 (GCLK) ของ EPM7128SLC84

- GCLK ต่ออยู่กับขา 83 ของ EPM7128SLC84

DC INPUT แรงดัน 7-10 Volt

เป็นแจ็คอแด็ปเตอร์ตัวเมียแกนเล็กสำหรับรับแรงดันไฟตรง 7 ถึง 10 โวลท์ ซึ่งภายในบอร์ดทดลองจะมีไอซี 7805 เร็คกูเรเตอร์แรงดันคงที่ 5 โวลท์ โดยที่แกนในจะเป็นขั้วบวก และแกนนอกเป็นขั้วลบ

ตำแหน่งขาของอุปกรณ์ต่างๆบนบอร์ด PLD - A01/1

ตารางที่ A1 ตำแหน่งขาของพอร์ตขยายช่องสัญญาณสำหรับ JP1

Hole Number	Signal/Pin	Hole Number	Signal/Pin
1	NO CONNECT	2	P11
3	P10	4	P9
5	P8	6	NO CONNECT
7	P6	8	P5
9	P4	10	NO CONNECT
11	OE2	12	GCLR
13	OE1	14	GCLK
15	NO CONNECT	16	P81
17	P80	18	P79
19	NO CONNECT	20	P77
21	P76	22	P75

ตารางที่ A2 ตำแหน่งขาของพอร์ตขยายช่องสัญญาณสำหรับ JP2

Hole Number	*Signal/Pin*	*Hole Number*	*Signal/Pin*
1	NO CONNECT	2	NO CONNECT
3	P31	4	P30
5	P29	6	P28
7	P27	8	NO CONNECT
9	P25	10	P24
11	TMS	12	P22
13	P21	14	P20
15	NO CONNECT	16	P18
17	P17	18	P16
19	P15	20	TDI
21	NO CONNECT	22	P12

ตารางที่ A3 ตำแหน่งขาของพอร์ตขยายช่องสัญญาณสำหรับ JP3

Hole Number	Signal/Pin	Hole Number	Signal/Pin
1	NO CONNECT	2	NO CONNECT
3	P52	4	P51
5	P50	6	P49
7	P48	8	NO CONNECT
9	P46	10	P45
11	P44	12	NO CONNECT
13	NO CONNECT	14	P41
15	P40	16	P39
17	NO CONNECT	18	P37
19	P36	20	P35
21	P34	22	P33

ตารางที่ A4 ตำแหน่งขาของพอร์ตขยายช่องสัญญาณสำหรับ JP4

Hole Number	Signal/Pin	Hole Number	Signal/Pin
1	NO CONNECT	2	P74
3	P73	4	NO CONNECT
5	TDO	6	P70
7	P69	8	P68
9	P67	10	NO CONNECT
11	P65	12	P64
13	P63	14	TCK
15	P61	16	P60
17	NO CONNECT	18	P58
19	P57	20	P56
21	P55	22	P54

*ตารางที่ A5 ตำแหน่งขาของสวิตช์แมตริกซ์ 4*4*

COLUMN*	PIN	ROW	PIN
1	28	1	22
2	29	2	24
3	30	3	25
4	31	4	27

ตารางที่ A6 ตำแหน่งขาของไดโอดเปล่งแสง

LED**	PIN	LED**	PIN
D0	21	D4	16
D1	20	D5	15
D2	18	D6	12
D3	17	D7	11

ตารางที่ A7 ตำแหน่งขาของ 7-Segment 4 หลัก

DISPLAY SEGMENT	PIN	DISPLAY SEGMENT	PIN
a	33	g	40
b	34	dot	41
c	35	C1	44
d	36	C2	45
e	37	C3	46
f	39	C4	48

ตารางที่ A8 ตำแหน่งขาของสวิตช์ กดติด – ปล่อยดับ

SWITCH*	1	2	3	4
PIN	49	50	51	52

ตารางที่ A9 ตำแหน่งขาของดิพสวิตช์ 8 หลัก

DIGIT*	PIN	DIGIT*	PIN
D0	54	D4	58
D1	55	D5	60
D2	56	D6	61
D3	57	D7	63

ตารางที่ A10 ตำแหน่งขาของ BUZZER และ OSCILLATOR

DEVICE	BUZZER	OSCILLATOR
PIN	10	83

หมายเหตุ : * มีตัวต้านทาน PULL UP 10K ต่ออยู่

** มีตัวต้านทาน 1K ต่ออนุกรมอยู่

ภาคผนวก ข.

โปรแกรมภาษาVHDL และวงจรที่ใช้งาน

โปรแกรมภาษาVHDL กำหนดสัญญาณ R, G และ B

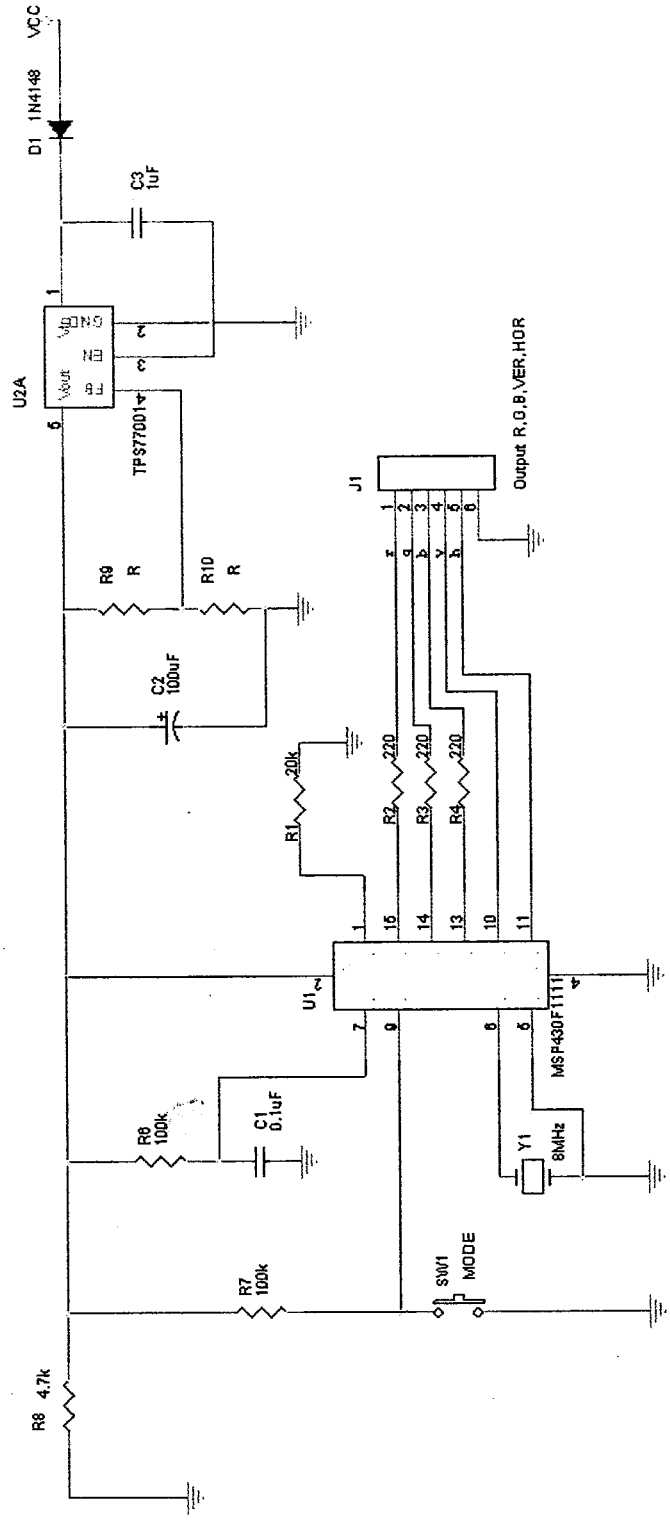
```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY rgbm IS
        PORT(C25 :      IN STD_LOGIC;
                r_in,g_in,b_in              : IN STD_LOGIC;
                hz_in,vz_in : IN STD_LOGIC ;
                sw : in std_logic_vector(5 downto 0);
                data : buffer std_logic_vector(5 downto 0);
                red_out,green_out,blue_out                  :       OUT STD_LOGIC;
           horiz_sync_out,vert_sync_out :          OUT STD_LOGIC);
END rgbm;

ARCHITECTURE RTL OF rgbm IS
        SIGNAL r_out,g_out,b_out        :       STD_LOGIC;
        SIGNAL rc                                       :       STD_LOGIC_VECTOR(9
DOWNTO 0);
        SIGNAL horiz_sync,vert_sync :  STD_LOGIC;
        --SIGNAL video_on,video_on_v,video_on_h : STD_LOGIC;
        --SIGNAL h_count,v_count     : STD_LOGIC_VECTOR(9 DOWNTO 0);
BEGIN
        horiz_sync_out<= hz_in;
        vert_sync_out<= vz_in;
        data <= sw;
        PROCESS(C25)
        BEGIN
        wait until (C25'event and C25='1');
```

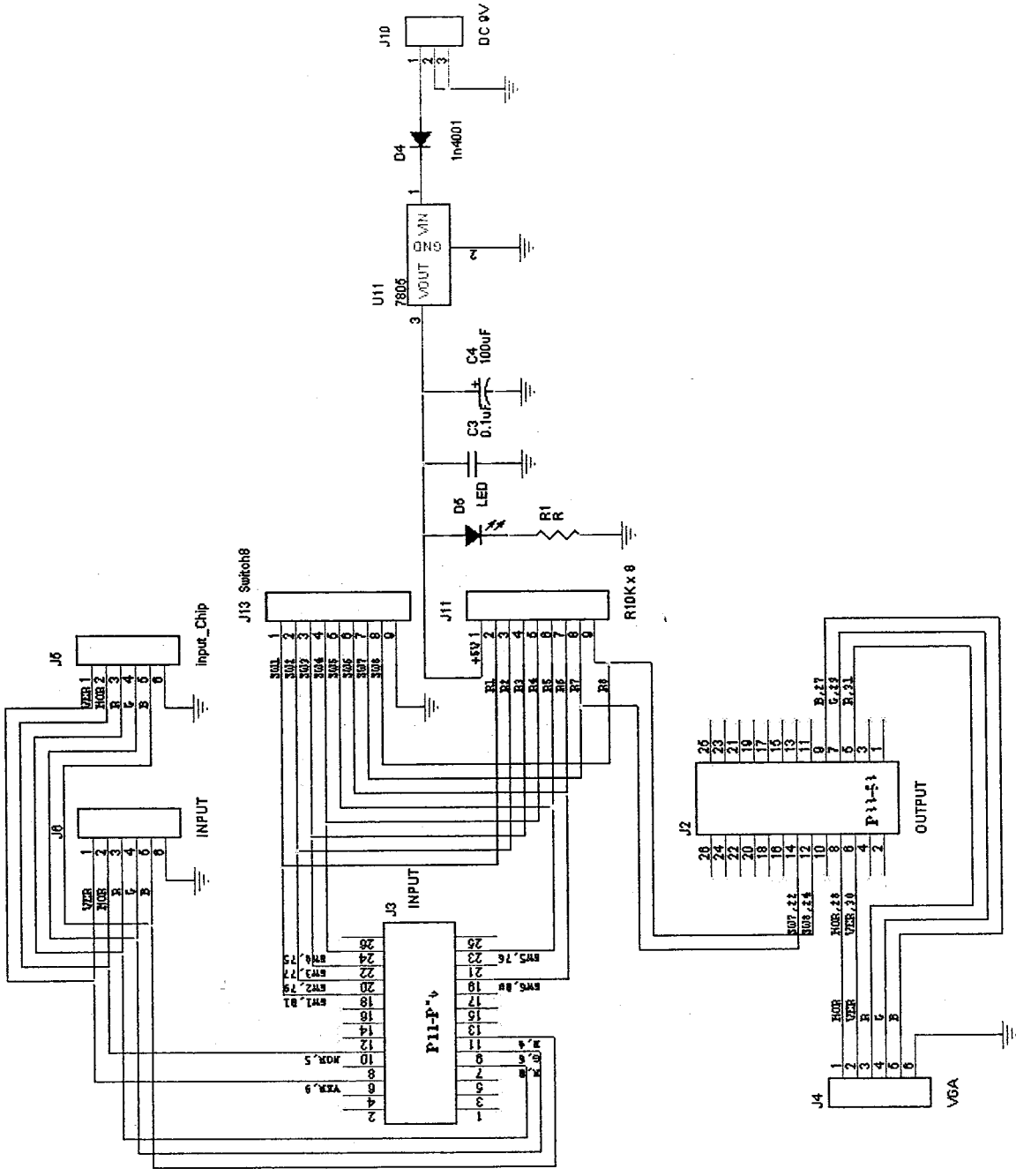
```
case DATA is
when "011111" =>
        r_out<=r_in;
        b_out<='0';
        g_out<='0';
when "101111" =>
        r_out<='0';--and hz_in;
        g_out<=g_in;
        b_out<='0';
when "110111" =>
        r_out<='0';
        g_out<='0';--and hz_in;
        b_out<=b_in;
when "111011" =>
        r_out<=r_in;
        g_out<=g_in;
        b_out<='0';--and hz_in;
when "111101" =>
        r_out<=r_in;
        g_out<='0';
        b_out<=b_in ;--and hz_in;
when "111110" =>
        r_out<='0';
        g_out<=g_in;
        b_out<=b_in ;--and hz_in;
when "111111" =>
        r_out<=r_in ;--and hz_in;
        g_out<=g_in ;--and hz_in;
        b_out<=b_in ;--and hz_in;
when "000000" =>
```

```
                r_out<=r_in ;--and hz_in;
                g_out<=g_in ;--and hz_in;
                b_out<=b_in ;--and hz_in;
        when others =>
                r_out<=r_in ;--and hz_in;
                g_out<=g_in ;--and hz_in;
                b_out<=b_in ;--and hz_in;
END case;
        --horiz_sync<=hz_in;
        --vert_sync<=vz_in;
        red_out<=r_out ;
        green_out<=g_out;
        blue_out<= b_out;
END PROCESS;
END RTL;
```

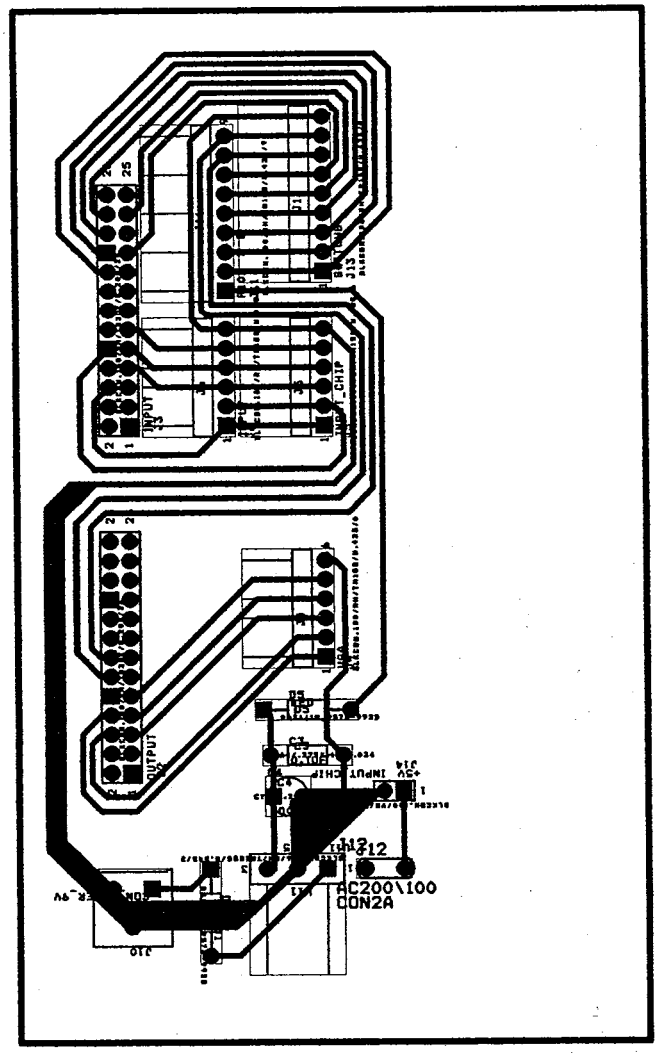
วงจรผลิตความถี่



วงจรรวมเชื่อมต่อระหว่างวงจรผลิตความถี่ไปยัง FPGA และจาก FPGA ไปยังเอาท์พุต

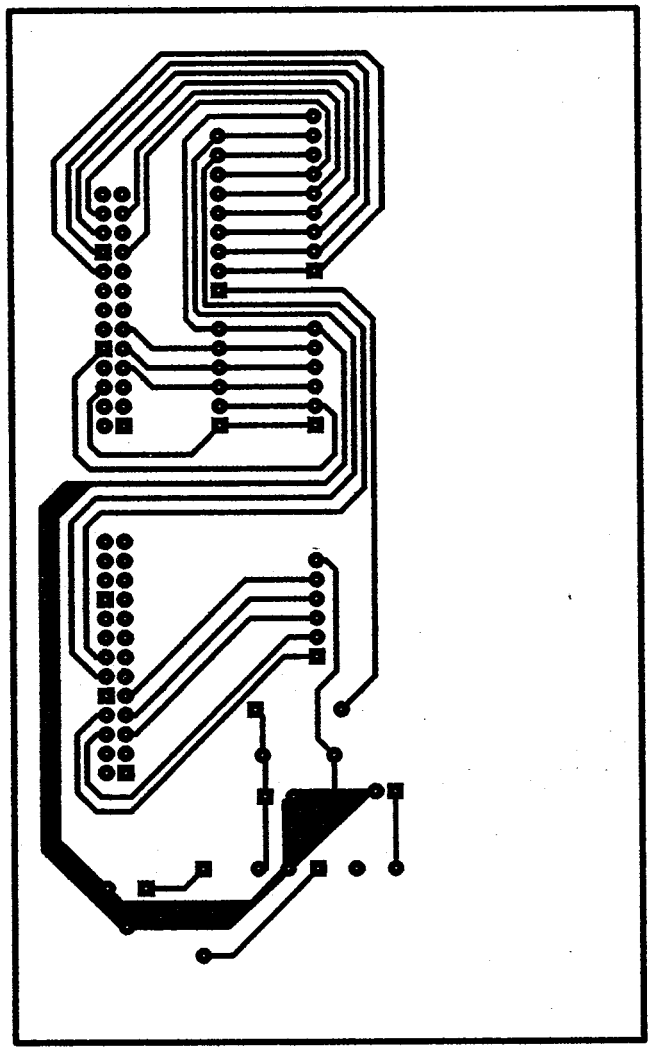


ลายปริ๊นซ์ และอุปกรณ์ของวงจรรวม

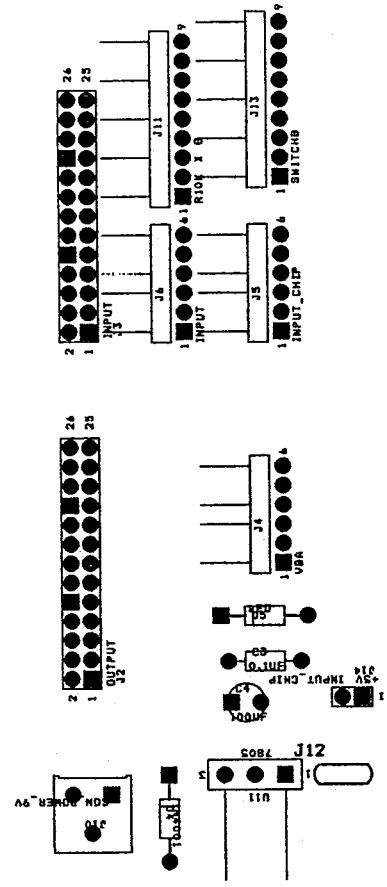


DRILL CHART				
SYM	DIAM	TOL	QTY	NOTE
x	0.085		106	
	TOTAL		106	

ลายปริ๊นซ์ของวงจรรวม



แสดงการวางอุปกรณ์ของวงจรรวม



ภาคผนวก ค.

ข้อมูล (Data Sheet)

Includes
MAX 7000E &
MAX 7000S

MAX 7000

Programmable Logic Device Family

August 2000, ver. 6.02 — Data Sheet

Features...

- High-performance, EEPROM-based programmable logic devices (PLDs) based on second-generation Multiple Array MatriX (MAX®) architecture
- 5.0-V in-system programmability (ISP) through the built-in IEEE Std. 1149.1 Joint Test Action Group (JTAG) interface available in MAX 7000S devices
- Includes 5.0-V MAX 7000 devices and 5.0-V ISP-based MAX 7000S devices
- Built-in JTAG boundary-scan test (BST) circuitry in MAX 7000S devices with 128 or more macrocells
- Complete EPLD family with logic densities ranging from 600 to 5,000 usable gates (see Tables 1 and 2)
- 5-ns pin-to-pin logic delays with up to 175.4-MHz counter frequencies (including interconnect)
- Peripheral component interconnect (PCI)-compliant devices available

For information on in-system programmable 3.3-V MAX 7000A or 2.5-V MAX 7000B devices, see the *MAX 7000A Programmable Logic Device Family Data Sheet* or the *MAX 7000B Programmable Logic Devices Advance Information Brief*.

Table 1. MAX 7000 Device Features

Feature	EPM7032	EPM7064	EPM7096	EPM7128E	EPM7160E	EPM7192E	EPM7256E
Usable gates	600	1,250	1,800	2,500	3,200	3,750	5,000
Macrocells	32	64	96	128	160	192	256
Logic array blocks	2	4	6	8	10	12	16
Maximum user I/O pins	36	68	76	100	104	124	164
t_{PD} (ns)	6	6	7.5	7.5	10	12	12
t_{SU} (ns)	5	5	6	6	7	7	7
t_{FSU} (ns)	2.5	2.5	3	3	3	3	3
t_{CO1} (ns)	4	4	4.5	4.5	5	6	6
f_{CNT} (MHz)	151.5	151.5	125.0	125.0	100.0	90.9	90.9

Table 2. MAX 7000S Device Features

Feature	EPM7032S	EPM7064S	EPM7128S	EPM7160S	EPM7192S	EPM7256S
Usable gates	600	1,250	2,500	3,200	3,750	5,000
Macrocells	32	64	128	160	192	256
Logic array blocks	2	4	8	10	12	16
Maximum user I/O pins	36	68	100	104	124	164
t_{PD} (ns)	5	5	6	6	7.5	7.5
t_{SU} (ns)	2.9	2.9	3.4	3.4	4.1	3.9
t_{FSU} (ns)	2.5	2.5	2.5	2.5	3	3
t_{CO1} (ns)	3.2	3.2	4	3.9	4.7	4.7
f_{CNT} (MHz)	175.4	175.4	147.1	149.3	125.0	128.2

...and More Features

- Open-drain output option in MAX 7000S devices
- Programmable macrocell flipflops with individual clear, preset, clock, and clock enable controls
- Programmable power-saving mode for a reduction of over 50% in each macrocell
- Configurable expander product-term distribution, allowing up to 32 product terms per macrocell
- 44 to 208 pins available in plastic J-lead chip carrier (PLCC), ceramic pin-grid array (PGA), plastic quad flat pack (PQFP), power quad flat pack (RQFP), and 1.0-mm thin quad flat pack (TQFP) packages
- Programmable security bit for protection of proprietary designs
- 3.3-V or 5.0-V operation
  - MultiVolt™ I/O interface operation, allowing devices to interface with 3.3-V or 5.0-V devices (MultiVolt I/O operation is not available in 44-pin packages)
  - Pin compatible with low-voltage MAX 7000A and MAX 7000B devices
- Enhanced features available in MAX 7000E and MAX 7000S devices
  - Six pin- or logic-driven output enable signals
  - Two global clock signals with optional inversion
  - Enhanced interconnect resources for improved routability
  - Fast input setup times provided by a dedicated path from I/O pin to macrocell registers
  - Programmable output slew-rate control
- Software design support and automatic place-and-route provided by Altera's MAX+PLUS® II development system for Windows-based PCs and Sun SPARCstation, HP 9000 Series 700/800, and IBM RISC System/6000 workstations, and the Quartus™ development system for Windows-based PCs and Sun SPARCstation and HP 9000 Series 700 workstations

- Additional design entry and simulation support provided by EDIF 2 0 0 and 3 0 0 netlist files, library of parameterized modules (LPM), Verilog HDL, VHDL, and other interfaces to popular EDA tools from manufacturers such as Cadence, Exemplar Logic, Mentor Graphics, OrCAD, Synopsys, and VeriBest
- Programming support
  - Altera's Master Programming Unit (MPU) and programming hardware from third-party manufacturers program all MAX 7000 devices
  - The BitBlaster™ serial download cable, ByteBlaster™ parallel port download cable, ByteBlasterMV™ parallel port download cable, and MasterBlaster™ serial/universal serial bus (USB) download cable program MAX 7000S devices

General Description

The MAX 7000 family of high-density, high-performance PLDs is based on Altera's second-generation MAX architecture. Fabricated with advanced CMOS technology, the EEPROM-based MAX 7000 family provides 600 to 5,000 usable gates, ISP, pin-to-pin delays as fast as 5 ns, and counter speeds of up to 175.4 MHz. MAX 7000S devices in the -5, -6, -7, and -10 speed grades as well as MAX 7000 and MAX 7000E devices in -5, -6, -7, -10P, and -12P speed grades comply with the PCI Special Interest Group (PCI SIG) ***PCI Local Bus Specification, Revision 2.2***. See Table 3 for available speed grades.

Table 3. MAX 7000 Speed Grades

Device	Speed Grade									
	-5	-6	-7	-10P	-10	-12P	-12	-15	-15T	-20
EPM7032		✓	✓		✓		✓	✓	✓	
EPM7032S	✓	✓	✓		✓					
EPM7064		✓	✓		✓		✓	✓		
EPM7064S	✓	✓	✓		✓					
EPM7096			✓		✓		✓	✓		
EPM7128E			✓	✓	✓		✓	✓		✓
EPM7128S		✓	✓		✓			✓		
EPM7160E				✓	✓		✓	✓		✓
EPM7160S		✓	✓		✓			✓		
EPM7192E						✓	✓	✓		✓
EPM7192S			✓		✓			✓		
EPM7256E						✓	✓	✓		✓
EPM7256S			✓		✓			✓		

The MAX 7000E devices—including the EPM7128E, EPM7160E, EPM7192E, and EPM7256E devices—have several enhanced features: additional global clocking, additional output enable controls, enhanced interconnect resources, fast input registers, and a programmable slew rate.

In-system programmable MAX 7000 devices—called MAX 7000S devices—include the EPM7032S, EPM7064S, EPM7128S, EPM7160S, EPM7192S, and EPM7256S devices. MAX 7000S devices have the enhanced features of MAX 7000E devices as well as JTAG BST circuitry in devices with 128 or more macrocells, ISP, and an open-drain output option. See Table 4.

Table 4. MAX 7000 Device Features

Feature	EPM7032 EPM7064 EPM7096	All MAX 7000E Devices	All MAX 7000S Devices
ISP via JTAG interface			✓
JTAG BST circuitry			✓ (1)
Open-drain output option			✓
Fast input registers		✓	✓
Six global output enables		✓	✓
Two global clocks		✓	✓
Slew-rate control		✓	✓
MultiVolt interface (2)	✓	✓	✓
Programmable register	✓	✓	✓
Parallel expanders	✓	✓	✓
Shared expanders	✓	✓	✓
Power-saving mode	✓	✓	✓
Security bit	✓	✓	✓
PCI-compliant devices available	✓	✓	✓

Notes:
(1) Available in EPM7128S, EPM7160S, EPM7192S, and EPM7256S devices only.
(2) The MultiVolt I/O interface is not available in 44-pin packages.

The MAX 7000 architecture supports 100% TTL emulation and high-density integration of SSI, MSI, and LSI logic functions. It easily integrates multiple devices ranging from PALs, GALs, and 22V10s to MACH and pLSI devices. MAX 7000 devices are available in a wide range of packages, including PLCC, PGA, PQFP, RQFP, and TQFP packages. See Table 5.

Table 5. MAX 7000 Maximum User I/O Pins *Note (1)*

Device	44-Pin PLCC	44-Pin PQFP	44-Pin TQFP	68-Pin PLCC	84-Pin PLCC	100-Pin PQFP	100-Pin TQFP	160-Pin PQFP	160-Pin PGA	192-Pin PGA	208-Pin PQFP	208-Pin RQFP
EPM7032	36	36	36									
EPM7032S	36		36									
EPM7064	36		36	52	68	68						
EPM7064S	36		36		68		68					
EPM7096				52	64	76						
EPM7128E					68	84		100				
EPM7128S					68	84	84 *(2)*	100				
EPM7160E					64	84		104				
EPM7160S					64		84 *(2)*	104				
EPM7192E								124	124			
EPM7192S								124				
EPM7256E								132 *(2)*		164		164
EPM7256S											164 *(2)*	164

Notes:
(1) When the JTAG interface in MAX 7000S devices is used, four I/O pins become JTAG pins.
(2) Perform a complete thermal analysis before committing a design to this device package. See the *Operating Requirements for Altera Devices Data Sheet* for more information.

MAX 7000 devices use CMOS EEPROM cells to implement logic functions. The user-configurable MAX 7000 architecture accommodates a variety of independent combinatorial and sequential logic functions. The devices can be reprogrammed for quick and efficient iterations during design development and debug cycles, and can be programmed and erased up to 100 times.

MAX 7000 devices contain from 32 to 256 macrocells that are combined into groups of 16 macrocells, called logic array blocks (LABs). Each macrocell has a programmable-AND/fixed-OR array and a configurable register with independently programmable clock, clock enable, clear, and preset functions. To build complex logic functions, each macrocell can be supplemented with both shareable expander product terms and high-speed parallel expander product terms to provide up to 32 product terms per macrocell.

The MAX 7000 family provides programmable speed/power optimization. Speed-critical portions of a design can run at high speed/full power, while the remaining portions run at reduced speed/low power. This speed/power optimization feature enables the designer to configure one or more macrocells to operate at 50% or lower power while adding only a nominal timing delay. MAX 7000E and MAX 7000S devices also provide an option that reduces the slew rate of the output buffers, minimizing noise transients when non-speed-critical signals are switching. The output drivers of all MAX 7000 devices (except 44-pin devices) can be set for either 3.3-V or 5.0-V operation, allowing MAX 7000 devices to be used in mixed-voltage systems.

The MAX 7000 family is supported by the Quartus and MAX+PLUS II development systems, a single, integrated package that allows schematic, text—including VHDL, Verilog HDL, and the Altera Hardware Description Language (AHDL)—and waveform design entry, compilation and logic synthesis, simulation and timing analysis, and device programming. The Quartus and MAX+PLUS II software provides EDIF 2 0 0 and 3 0 0, LPM, VHDL, Verilog HDL, and other interfaces for additional design entry and simulation support from other industry-standard PC- and UNIX-workstation-based EDA tools. The MAX+PLUS II software runs on Windows-based PCs, as well as Sun SPARCstation, HP 9000 Series 700/800, and IBM RISC System/6000 workstations. The Quartus software runs on Windows-based PCs, as well as Sun SPARCstation and HP 9000 Series 700 workstations.

For more information on development tools, go to the *MAX+PLUS II Programmable Logic Development System & Software Data Sheet*.

Functional Description

The MAX 7000 architecture includes the following elements:

- Logic array blocks
- Macrocells
- Expander product terms (shareable and parallel)
- Programmable interconnect array
- I/O control blocks

The MAX 7000 architecture includes four dedicated inputs that can be used as general-purpose inputs or as high-speed, global control signals (clock, clear, and two output enable signals) for each macrocell and I/O pin. Figure 1 shows the architecture of EPM7032, EPM7064, and EPM7096 devices.

Figure 1. EPM7032, EPM7064 & EPM7096 Device Block Diagram

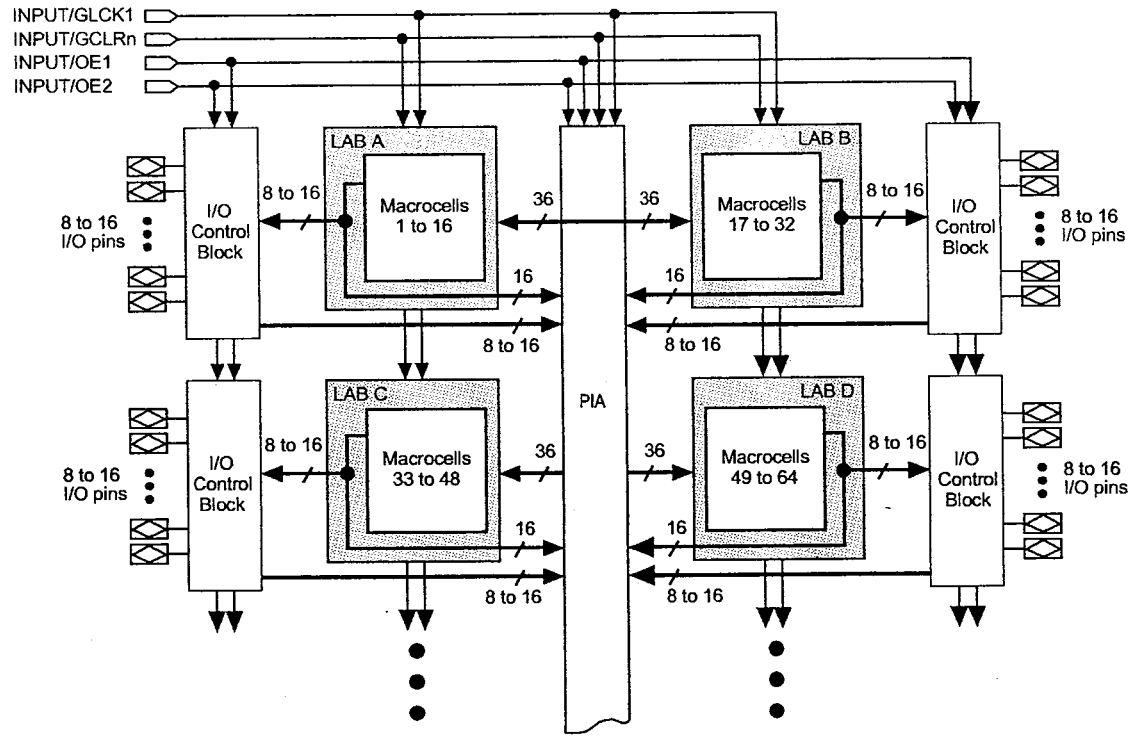
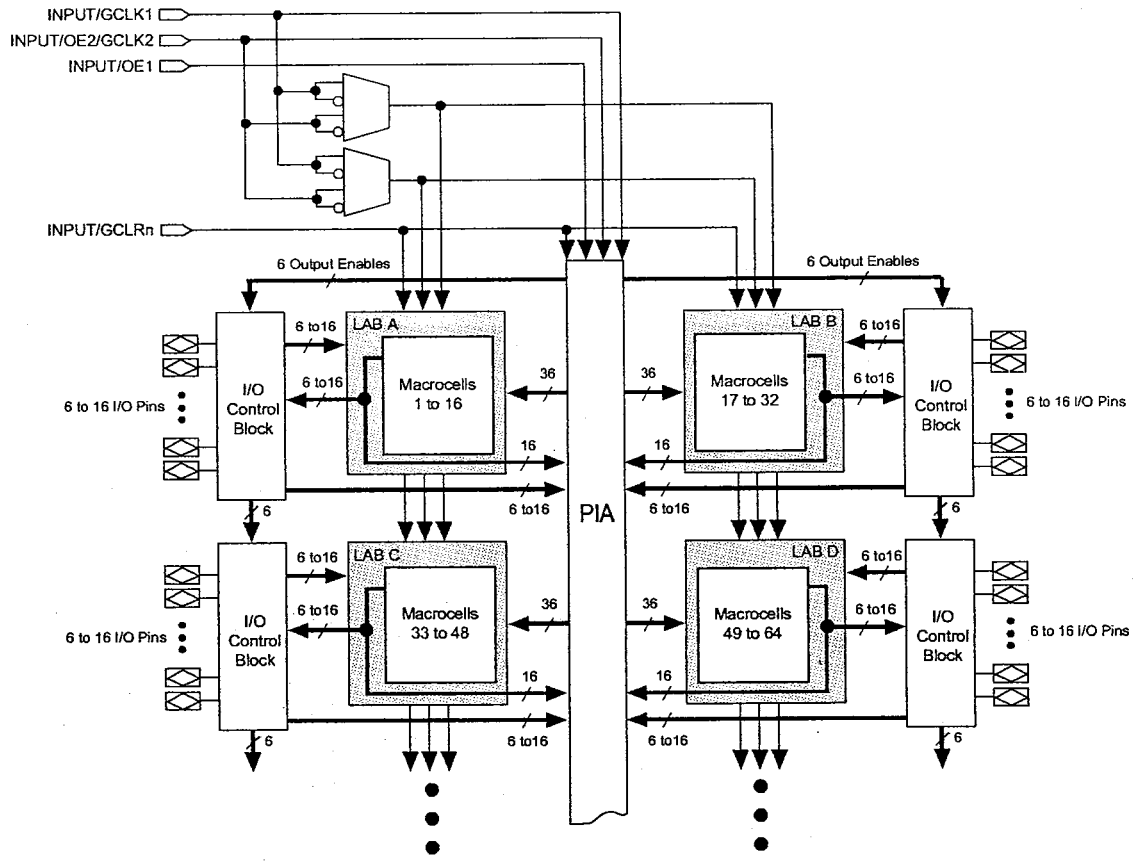


Figure 2 shows the architecture of MAX 7000E and MAX 7000S devices.

Figure 2. MAX 7000E & MAX 7000S Device Block Diagram



Logic Array Blocks

The MAX 7000 device architecture is based on the linking of high-performance, flexible, logic array modules called logic array blocks (LABs). LABs consist of 16-macrocell arrays, as shown in Figures 1 and 2. Multiple LABs are linked together via the programmable interconnect array (PIA), a global bus that is fed by all dedicated inputs, I/O pins, and macrocells.

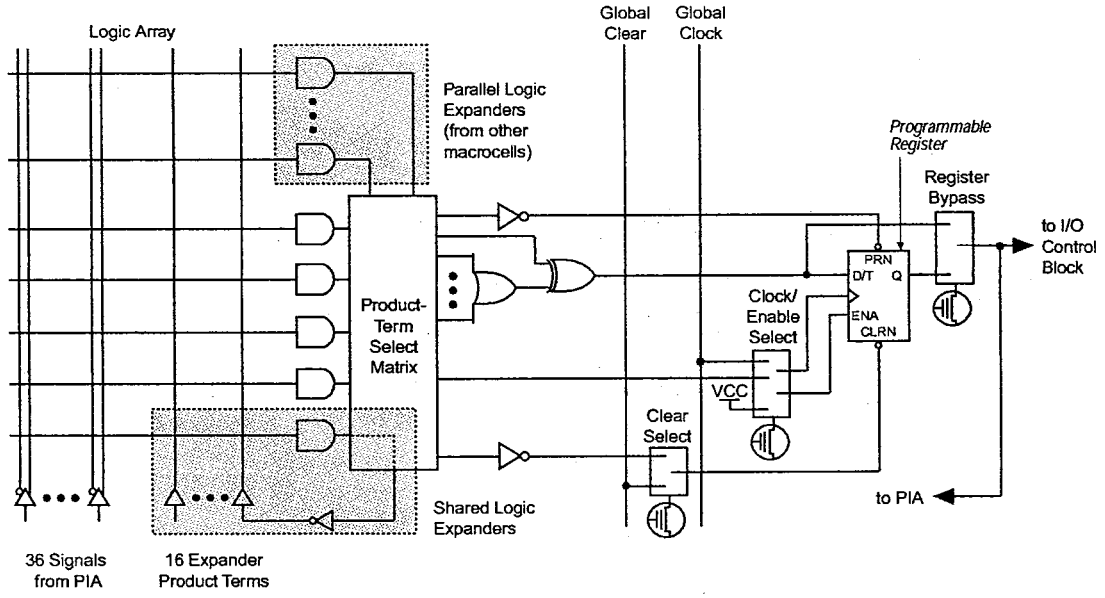
Each LAB is fed by the following signals:

- 36 signals from the PIA that are used for general logic inputs
- Global controls that are used for secondary register functions
- Direct input paths from I/O pins to the registers that are used for fast setup times for MAX 7000E and MAX 7000S devices

Macrocells

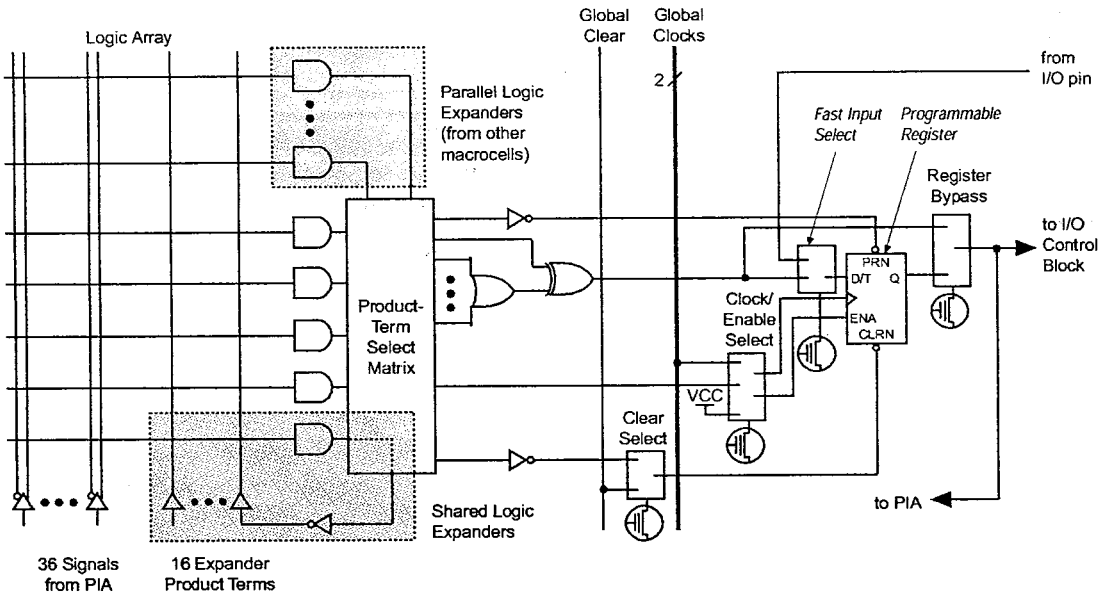
The MAX 7000 macrocell can be individually configured for either sequential or combinatorial logic operation. The macrocell consists of three functional blocks: the logic array, the product-term select matrix, and the programmable register. The macrocell of EPM7032, EPM7064, and EPM7096 devices is shown in Figure 3.

Figure 3. EPM7032, EPM7064 & EPM7096 Device Macrocell



The macrocell of MAX 7000E and MAX 7000S devices is shown in Figure 4.

Figure 4. MAX 7000E & MAX 7000S Device Macrocell



Combinatorial logic is implemented in the logic array, which provides five product terms per macrocell. The product-term select matrix allocates these product terms for use as either primary logic inputs (to the OR and XOR gates) to implement combinatorial functions, or as secondary inputs to the macrocell's register clear, preset, clock, and clock enable control functions. Two kinds of expander product terms ("expanders") are available to supplement macrocell logic resources:

- Shareable expanders, which are inverted product terms that are fed back into the logic array
- Parallel expanders, which are product terms borrowed from adjacent macrocells

The Quartus and MAX+PLUS II software automatically optimizes product-term allocation according to the logic requirements of the design.

For registered functions, each macrocell flipflop can be individually programmed to implement D, T, JK, or SR operation with programmable clock control. The flipflop can be bypassed for combinatorial operation. During design entry, the designer specifies the desired flipflop type; the Quartus and MAX+PLUS II software then selects the most efficient flipflop operation for each registered function to optimize resource utilization.

Each programmable register can be clocked in three different modes:

- By a global clock signal. This mode achieves the fastest clock-to-output performance.
- By a global clock signal and enabled by an active-high clock enable. This mode provides an enable on each flipflop while still achieving the fast clock-to-output performance of the global clock.
- By an array clock implemented with a product term. In this mode, the flipflop can be clocked by signals from buried macrocells or I/O pins.

In EPM7032, EPM7064, and EPM7096 devices, the global clock signal is available from a dedicated clock pin, GCLK1, as shown in Figure 1. In MAX 7000E and MAX 7000S devices, two global clock signals are available. As shown in Figure 2, these global clock signals can be the true or the complement of either of the global clock pins, GCLK1 or GCLK2.

Each register also supports asynchronous preset and clear functions. As shown in Figures 3 and 4, the product-term select matrix allocates product terms to control these operations. Although the product-term-driven preset and clear of the register are active high, active-low control can be obtained by inverting the signal within the logic array. In addition, each register clear function can be individually driven by the active-low dedicated global clear pin (GCLRn).

All MAX 7000E and MAX 7000S I/O pins have a fast input path to a macrocell register. This dedicated path allows a signal to bypass the PIA and combinatorial logic and be driven to an input D flipflop with an extremely fast (2.5-ns) input setup time.

Expander Product Terms

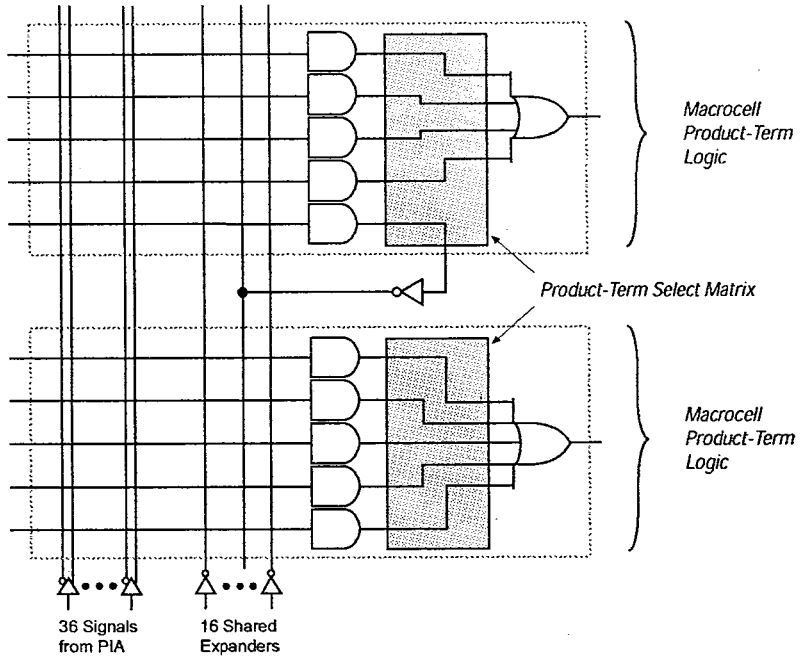
Although most logic functions can be implemented with the five product terms available in each macrocell, the more complex logic functions require additional product terms. Another macrocell can be used to supply the required logic resources; however, the MAX 7000 architecture also allows both shareable and parallel expander product terms ("expanders") that provide additional product terms directly to any macrocell in the same LAB. These expanders help ensure that logic is synthesized with the fewest possible logic resources to obtain the fastest possible speed.

Shareable Expanders

Each LAB has 16 shareable expanders that can be viewed as a pool of uncommitted single product terms (one from each macrocell) with inverted outputs that feed back into the logic array. Each shareable expander can be used and shared by any or all macrocells in the LAB to build complex logic functions. A small delay (t_{SEXP}) is incurred when shareable expanders are used. Figure 5 shows how shareable expanders can feed multiple macrocells.

Figure 5. Shareable Expanders

Shareable expanders can be shared by any or all macrocells in an LAB.



Parallel Expanders

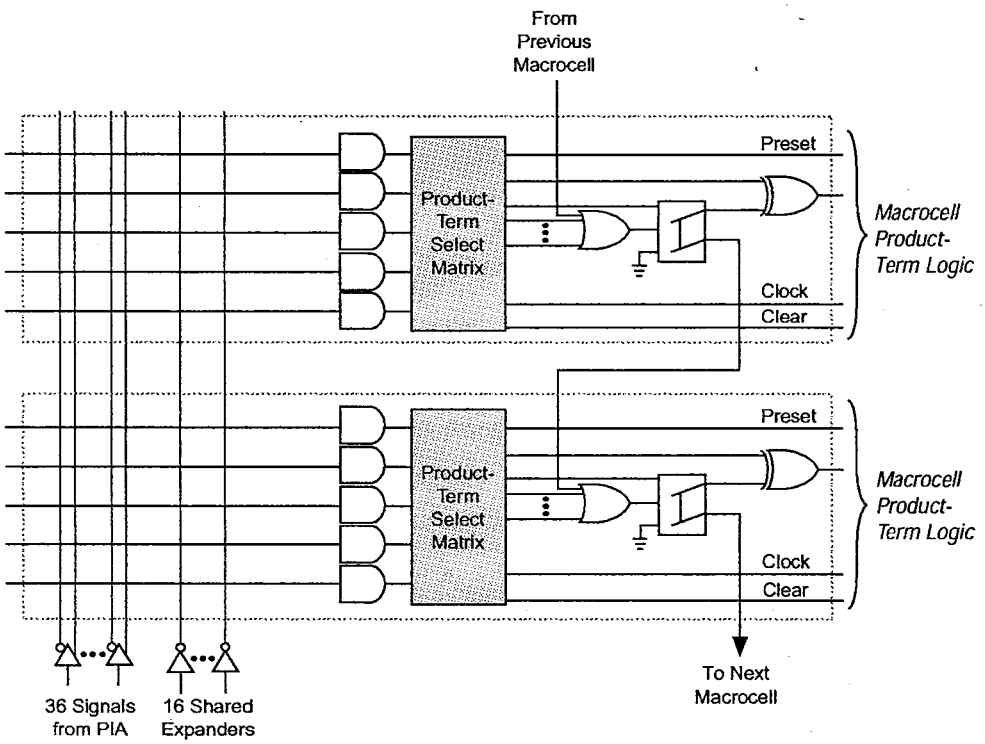
Parallel expanders are unused product terms that can be allocated to a neighboring macrocell to implement fast, complex logic functions. Parallel expanders allow up to 20 product terms to directly feed the macrocell OR logic, with five product terms provided by the macrocell and 15 parallel expanders provided by neighboring macrocells in the LAB.

The Quartus and MAX+PLUS II Compilers can allocate up to three sets of up to five parallel expanders automatically to the macrocells that require additional product terms. Each set of five parallel expanders incurs a small, incremental timing delay (t_{PEXP}). For example, if a macrocell requires 14 product terms, the Compiler uses the five dedicated product terms within the macrocell and allocates two sets of parallel expanders; the first set includes five product terms and the second set includes 4 product terms, increasing the total delay by $2 \times t_{PEXP}$.

Two groups of 8 macrocells within each LAB (e.g., macrocells 1 through 8 and 9 through 16) form two chains to lend or borrow parallel expanders. A macrocell borrows parallel expanders from lower-numbered macrocells. For example, macrocell 8 can borrow parallel expanders from macrocell 7, from macrocells 7 and 6, or from macrocells 7, 6, and 5. Within each group of 8, the lowest-numbered macrocell can only lend parallel expanders and the highest-numbered macrocell can only borrow them. Figure 6 shows how parallel expanders can be borrowed from a neighboring macrocell.

Figure 6. Parallel Expanders

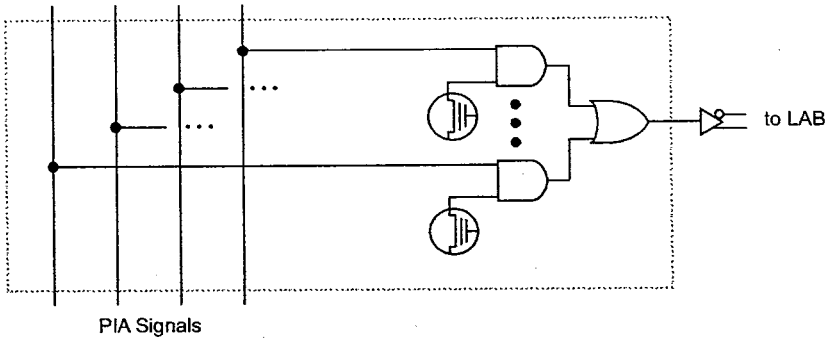
Unused product terms in a macrocell can be allocated to a neighboring macrocell.



Programmable Interconnect Array

Logic is routed between LABs via the programmable interconnect array (PIA). This global bus is a programmable path that connects any signal source to any destination on the device. All MAX 7000 dedicated inputs, I/O pins, and macrocell outputs feed the PIA, which makes the signals available throughout the entire device. Only the signals required by each LAB are actually routed from the PIA into the LAB. Figure 7 shows how the PIA signals are routed into the LAB. An EEPROM cell controls one input to a 2-input AND gate, which selects a PIA signal to drive into the LAB.

Figure 7. PIA Routing



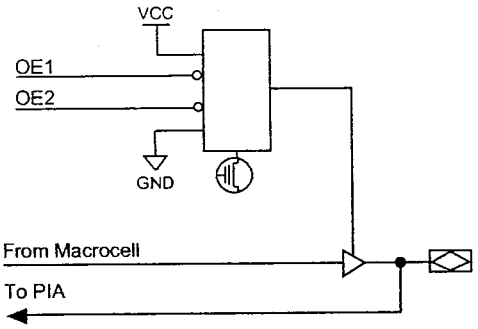
While the routing delays of channel-based routing schemes in masked or field-programmable gate arrays (FPGAs) are cumulative, variable, and path-dependent, the MAX 7000 PIA has a fixed delay. The PIA thus eliminates skew between signals and makes timing performance easy to predict.

I/O Control Blocks

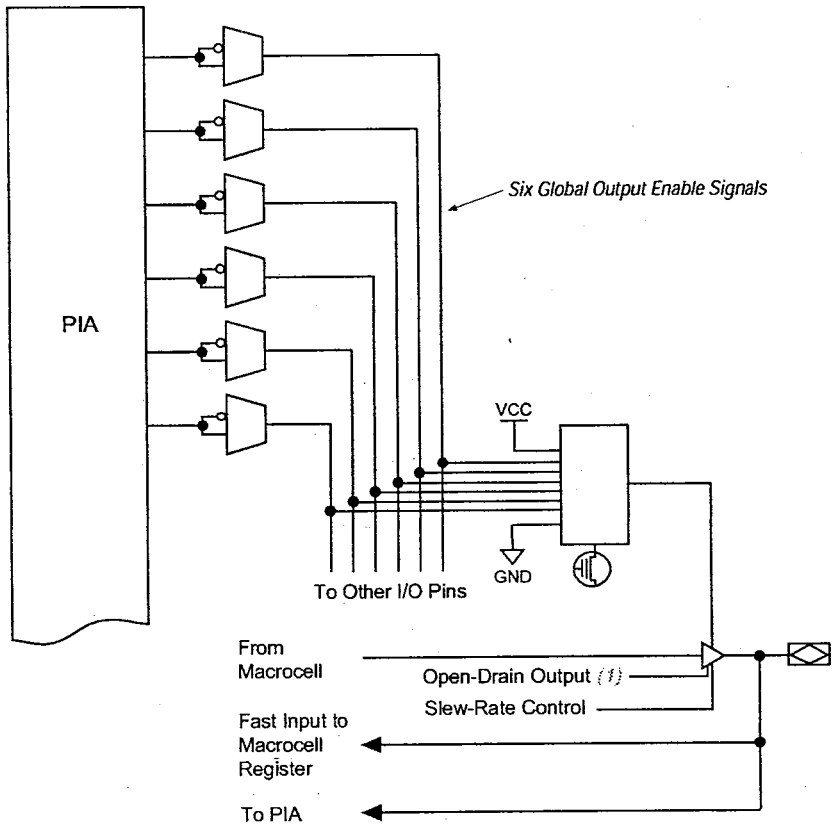
The I/O control block allows each I/O pin to be individually configured for input, output, or bidirectional operation. All I/O pins have a tri-state buffer that is individually controlled by one of the global output enable signals or directly connected to ground or V_{CC}. Figure 8 shows the I/O control block for the MAX 7000 family. The I/O control block of EPM7032, EPM7064, and EPM7096 devices has two global output enable signals that are driven by two dedicated active-low output enable pins (OE1 and OE2). The I/O control block of MAX 7000E and MAX 7000S devices has six global output enable signals that are driven by the true or complement of two output enable signals, a subset of the I/O pins, or a subset of the I/O macrocells.

Figure 8. I/O Control Block of MAX 7000 Devices

EPM7032, EPM7064 & EPM7096 Devices



MAX 7000E & MAX 7000S Devices



Note:
(1) The open-drain output option is available in MAX 7000S devices only.

When the tri-state buffer control is connected to ground, the output is tri-stated (high impedance) and the I/O pin can be used as a dedicated input. When the tri-state buffer control is connected to V_{CC}, the output is enabled.

The MAX 7000 architecture provides dual I/O feedback, in which macrocell and pin feedbacks are independent. When an I/O pin is configured as an input, the associated macrocell can be used for buried logic.

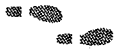
In-System Programmability (ISP)

MAX 7000S devices are in-system programmable via an industry-standard 4-pin Joint Test Action Group (JTAG) interface (IEEE Std. 1149.1-1990). ISP allows quick, efficient iterations during design development and debugging cycles. The MAX 7000S architecture internally generates the high programming voltage required to program EEPROM cells, allowing in-system programming with only a single 5.0 V power supply. During in-system programming, the I/O pins are tri-stated and pulled-up to eliminate board conflicts. The pull-up value is nominally 50 kΩ.

ISP simplifies the manufacturing flow by allowing devices to be mounted on a printed circuit board with standard in-circuit test equipment before they are programmed. MAX 7000S devices can be programmed by downloading the information via in-circuit testers (ICT), embedded processors, or the Altera BitBlaster, ByteBlaster, ByteBlasterMV, or MasterBlaster download cables. (The ByteBlaster cable is obsolete and is replaced by the ByteBlasterMV cable, which can program and configure 2.5-V, 3.3-V, and 5.0-V devices.) Programming the devices after they are placed on the board eliminates lead damage on high-pin-count packages (e.g., QFP packages) due to device handling and allows devices to be reprogrammed after a system has already shipped to the field. For example, product upgrades can be performed in the field via software or modem.

In-system programming can be accomplished with either an adaptive or constant algorithm. An adaptive algorithm reads information from the unit and adapts subsequent programming steps to achieve the fastest possible programming time for that unit. Because some in-circuit testers can not support an adaptive algorithm, Altera offers devices tested with a constant algorithm. Devices tested to the constant algorithm are marked with an "F" suffix in the ordering code.

The Jam™ programming and test language can be used to program MAX 7000S devices with in-circuit test equipment (e.g., PC, embedded processor).



For more information on using the Jam language, see *Application Note 88 (Using the Jam Language for ISP & ICR via an Embedded Processor)*.

Programmable Speed/Power Control

MAX 7000 devices offer a power-saving mode that supports low-power operation across user-defined signal paths or the entire device. This feature allows total power dissipation to be reduced by 50% or more, because most logic applications require only a small fraction of all gates to operate at maximum frequency.

The designer can program each individual macrocell in a MAX 7000 device for either high-speed (i.e., with the Turbo Bit™ option turned on) or low-power (i.e., with the Turbo Bit option turned off) operation. As a result, speed-critical paths in the design can run at high speed, while the remaining paths can operate at reduced power. Macrocells that run at low power incur a nominal timing delay adder (t_{LPA}) for the t_{LAD}, t_{LAC}, t_{IC}, t_{EN}, and t_{SEXP}, t_{ACL}, and t_{CPPW} parameters.

Output Configuration

MAX 7000 device outputs can be programmed to meet a variety of system-level requirements.

MultiVolt I/O Interface

MAX 7000 devices—except 44-pin devices—support the MultiVolt I/O interface feature, which allows MAX 7000 devices to interface with systems that have differing supply voltages. The 5.0-V devices in all packages can be set for 3.3-V or 5.0-V I/O pin operation. These devices have one set of VCC pins for internal operation and input buffers (VCCINT), and another set for I/O output drivers (VCCIO).

The VCCINT pins must always be connected to a 5.0-V power supply. With a 5.0-V V_{CCINT} level, input voltage thresholds are at TTL levels, and are therefore compatible with both 3.3-V and 5.0-V inputs.

The VCCIO pins can be connected to either a 3.3-V or a 5.0-V power supply, depending on the output requirements. When the VCCIO pins are connected to a 5.0-V supply, the output levels are compatible with 5.0-V systems. When V_{CCIO} is connected to a 3.3-V supply, the output high is 3.3 V and is therefore compatible with 3.3-V or 5.0-V systems. Devices operating with V_{CCIO} levels lower than 4.75 V incur a nominally greater timing delay of t_{OD2} instead of t_{OD1}.

Open-Drain Output Option (MAX 7000S Devices Only)

MAX 7000S devices provide an optional open-drain (functionally equivalent to open-collector) output for each I/O pin. This open-drain output enables the device to provide system-level control signals (e.g., interrupt and write enable signals) that can be asserted by any of several devices. It can also provide an additional wired-OR plane.

Output pins on 5.0-V MAX 7000S devices with V_{CCIO} = 3.3 V or 5.0 V (with a pull-up resistor to the 5.0-V supply) can also drive 5.0-V CMOS input pins. In this case, the pull-up transistor will turn off when the pin voltage exceeds 3.3 V. Therefore, the pin does not have to be open-drain.

Slew-Rate Control

The output buffer for each MAX 7000E and MAX 7000S I/O pin has an adjustable output slew rate that can be configured for low-noise or high-speed performance. A faster slew rate provides high-speed transitions for high-performance systems. However, these fast transitions may introduce noise transients into the system. A slow slew rate reduces system noise, but adds a nominal delay of 4 to 5 ns. In MAX 7000E devices, when the Turbo Bit is turned off, the slew rate is set for low noise performance. For MAX 7000S devices, each I/O pin has an individual EEPROM bit that controls the slew rate, allowing designers to specify the slew rate on a pin-by-pin basis.

Programming with External Hardware

MAX 7000 devices can be programmed on Windows-based PCs with the MAX+PLUS II Programmer, an Altera Logic Programmer card, the Master Programming Unit (MPU), and the appropriate device adapter. The MPU performs a continuity check to ensure adequate electrical contact between the adapter and the device. For more information, see the *Altera Programming Hardware Data Sheet*.

The MAX+PLUS II software can use text- or waveform-format test vectors created with the MAX+PLUS II Text Editor or Waveform Editor to test the programmed device. For added design verification, designers can perform functional testing to compare the functional behavior of a MAX 7000 device with the results of simulation. Moreover, Data I/O, BP Microsystems, and other programming hardware manufacturers also provide programming support for Altera devices. For more information, see *Programming Hardware Manufacturers*.

IEEE Std. 1149.1 (JTAG) Boundary-Scan Support

MAX 7000 devices support JTAG BST circuitry as specified by IEEE Std. 1149.1-1990. Table 6 describes the JTAG instructions supported by the MAX 7000 family. The pin-out tables starting on page 55 of this data sheet show the location of the JTAG control pins for each device. If the JTAG interface is not required, the JTAG pins are available as user I/O pins.

Table 6. MAX 7000 JTAG Instructions

JTAG Instruction	Devices	Description
SAMPLE/PRELOAD	EPM7128S EPM7160S EPM7192S EPM7256S	Allows a snapshot of signals at the device pins to be captured and examined during normal device operation, and permits an initial data pattern output at the device pins.
EXTEST	EPM7128S EPM7160S EPM7192S EPM7256S	Allows the external circuitry and board-level interconnections to be tested by forcing a test pattern at the output pins and capturing test results at the input pins.
BYPASS	EPM7032S EPM7064S EPM7128S EPM7160S EPM7192S EPM7256S	Places the 1-bit bypass register between the TDI and TDO pins, which allows the BST data to pass synchronously through a selected device to adjacent devices during normal device operation.
IDCODE	EPM7032S EPM7064S EPM7128S EPM7160S EPM7192S EPM7256S	Selects the IDCODE register and places it between TDI and TDO, allowing the IDCODE to be serially shifted out of TDO.
ISP Instructions	EPM7032S EPM7064S EPM7128S EPM7160S EPM7192S EPM7256S	These instructions are used when programming MAX 7000S devices via the JTAG ports with the BitBlaster, ByteBlaster, ByteBlasterMV, or MasterBlaster download cable, or using a Jam File (.jam), Jam Byte-Code (.jbc), or Serial Vector Format (.svf) file via an embedded processor or test equipment.

The instruction register length of MAX 7000S devices is 10 bits. Tables 7 and 8 show the boundary-scan register length and device IDCODE information for MAX 7000S devices.

Table 7. MAX 7000S Boundary-Scan Register Length

Device	Boundary-Scan Register Length
EPM7032S	1 (1)
EPM7064S	1 (1)
EPM7128S	288
EPM7160S	312
EPM7192S	360
EPM7256S	480

Note:
(1) This device does not support JTAG boundary-scan testing.

Table 8. 32-Bit MAX 7000 Device IDCODE *Note (1)*

Device	IDCODE (32 Bits)			
	Version (4 Bits)	Part Number (16 Bits)	Manufacturer's Identity (11 Bits)	1 (1 Bit) (2)
EPM7032S	0000	0111 0000 0011 0010	00001101110	1
EPM7064S	0000	0111 0000 0110 0100	00001101110	1
EPM7128S	0000	0111 0001 0010 1000	00001101110	1
EPM7160S	0000	0111 0001 0110 0000	00001101110	1
EPM7192S	0000	0111 0001 1001 0010	00001101110	1
EPM7256S	0000	0111 0010 0101 0110	00001101110	1

Notes:
(1) The most significant bit (MSB) is on the left.
(2) The least significant bit (LSB) for all JTAG IDCODEs is 1.

Figure 9 shows the timing requirements for the JTAG signals.

Figure 9. MAX 7000 JTAG Waveforms

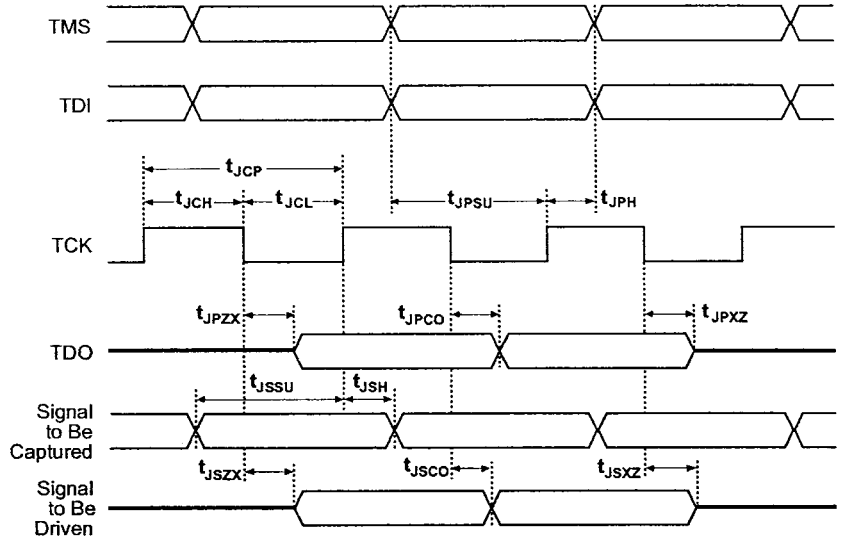


Table 9 shows the JTAG timing parameters and values for MAX 7000S devices.

Table 9. JTAG Timing Parameters & Values for MAX 7000S Devices

Symbol	Parameter	Min	Max	Unit
t_{JCP}	TCK clock period	100		ns
t_{JCH}	TCK clock high time	50		ns
t_{JCL}	TCK clock low time	50		ns
t_{JPSU}	JTAG port setup time	20		ns
t_{JPH}	JTAG port hold time	45		ns
t_{JPCO}	JTAG port clock to output		25	ns
t_{JPZX}	JTAG port high impedance to valid output		25	ns
t_{JPXZ}	JTAG port valid output to high impedance		25	ns
t_{JSSU}	Capture register setup time	20		ns
t_{JSH}	Capture register hold time	45		ns
t_{JSCO}	Update register clock to output		25	ns
t_{JSZX}	Update register high impedance to valid output		25	ns
t_{JSXZ}	Update register valid output to high impedance		25	ns

For more information, see *Application Note 39 (IEEE 1149.1 (JTAG) Boundary-Scan Testing in Altera Devices)*.

Design Security

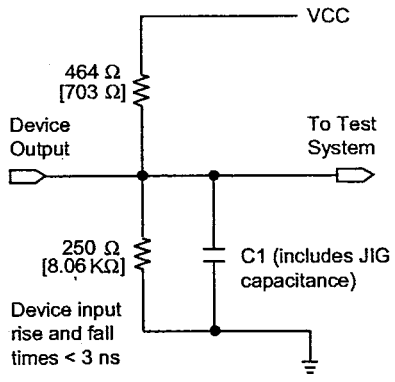
All MAX 7000 devices contain a programmable security bit that controls access to the data programmed into the device. When this bit is programmed, a proprietary design implemented in the device cannot be copied or retrieved. This feature provides a high level of design security, because programmed data within EEPROM cells is invisible. The security bit that controls this function, as well as all other programmed data, is reset only when the device is reprogrammed.

Generic Testing

Each MAX 7000 device is functionally tested. Complete testing of each programmable EEPROM bit and all internal logic elements ensures 100% programming yield. AC test measurements are taken under conditions equivalent to those shown in Figure 10. Test patterns can be used and then erased during early stages of the production flow.

Figure 10. MAX 7000 AC Test Conditions

Power supply transients can affect AC measurements. Simultaneous transitions of multiple outputs should be avoided for accurate measurement. Threshold tests must not be performed under AC conditions. Large-amplitude, fast ground-current transients normally occur as the device outputs discharge the load capacitances. When these transients flow through the parasitic inductance between the device ground pin and the test system ground, significant reductions in observable noise immunity can result. Numbers in brackets are for 2.5-V devices and outputs. Numbers without brackets are for 3.3-V devices and outputs.



QFP Carrier & Development Socket

MAX 7000 and MAX 7000E devices in QFP packages with 100 or more pins are shipped in special plastic carriers to protect the QFP leads. The carrier is used with a prototype development socket and special programming hardware available from Altera. This carrier technology makes it possible to program, test, erase, and reprogram a device without exposing the leads to mechanical stress.

For detailed information and carrier dimensions, refer to the *QFP Carrier & Development Socket Data Sheet*.

MAX 7000S devices are not shipped in carriers.

Operating Conditions

Tables 10 through 15 provide information about absolute maximum ratings, recommended operating conditions, operating conditions, and capacitance for 5.0-V MAX 7000 devices.

Table 10. MAX 7000 5.0-V Device Absolute Maximum Ratings *Note (1)*

Symbol	Parameter	Conditions	Min	Max	Unit
V_{CC}	Supply voltage	With respect to ground *(2)*	–2.0	7.0	V
V_I	DC input voltage		–2.0	7.0	V
I_{OUT}	DC output current, per pin		–25	25	mA
T_{STG}	Storage temperature	No bias	–65	150	° C
T_{AMB}	Ambient temperature	Under bias	–65	135	° C
T_J	Junction temperature	Ceramic packages, under bias		150	° C
		PQFP and RQFP packages, under bias		135	° C

Table 11. MAX 7000 5.0-V Device Recommended Operating Conditions

Symbol	Parameter	Conditions	Min	Max	Unit
V_{CCINT}	Supply voltage for internal logic and input buffers	*(3)*, *(4)*	4.75 (4.50)	5.25 (5.50)	V
V_{CCIO}	Supply voltage for output drivers, 5.0-V operation	*(3)*, *(4)*	4.75 (4.50)	5.25 (5.50)	V
	Supply voltage for output drivers, 3.3-V operation	*(3)*, *(4)*, *(5)*	3.00 (3.00)	3.60 (3.60)	V
V_{CCISP}	Supply voltage during ISP	*(6)*	4.75	5.25	V
V_I	Input voltage		–0.5 *(7)*	V_{CCINT} + 0.5	V
V_O	Output voltage		0	V_{CCIO}	V
T_A	Ambient temperature	For commercial use	0	70	° C
		For industrial use	–40	85	° C
T_J	Junction temperature	For commercial use	0	90	° C
		For industrial use	–40	105	° C
t_R	Input rise time			40	ns
t_F	Input fall time			40	ns

Table 12. MAX 7000 5.0-V Device DC Operating Conditions *Note (8)*

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IH}	High-level input voltage		2.0	V_{CCINT} + 0.5	V
V_{IL}	Low-level input voltage		–0.5 *(7)*	0.8	V
V_{OH}	5.0-V high-level TTL output voltage	I_{OH} = –4 mA DC, V_{CCIO} = 4.75 V *(9)*	2.4		V
	3.3-V high-level TTL output voltage	I_{OH} = –4 mA DC, V_{CCIO} = 3.00 V *(9)*	2.4		V
	3.3-V high-level CMOS output voltage	I_{OH} = –0.1 mA DC, V_{CCIO} = 3.0 V *(9)*	V_{CCIO} – 0.2		V
V_{OL}	5.0-V low-level TTL output voltage	I_{OL} = 12 mA DC, V_{CCIO} = 4.75 V *(10)*		0.45	V
	3.3-V low-level TTL output voltage	I_{OL} = 12 mA DC, V_{CCIO} = 3.00 V *(10)*		0.45	V
	3.3-V low-level CMOS output voltage	I_{OL} = 0.1 mA DC, V_{CCIO} = 3.0 V *(10)*		0.2	V
I_I	Leakage current of dedicated input pins	V_I = V_{CC} or ground	–10	10	µA
I_{OZ}	I/O pin tri-state output off-state current	V_O = V_{CC} or ground *(11)*	–40	40	µA

Table 13. MAX 7000 5.0-V Device Capacitance: EPM7032, EPM7064 & EPM7096 Devices *Note (12)*

Symbol	Parameter	Conditions	Min	Max	Unit
C_{IN}	Input pin capacitance	V_{IN} = 0 V, f = 1.0 MHz		12	pF
$C_{I/O}$	I/O pin capacitance	V_{OUT} = 0 V, f = 1.0 MHz		12	pF

Table 14. MAX 7000 5.0-V Device Capacitance: MAX 7000E Devices *Note (12)*

Symbol	Parameter	Conditions	Min	Max	Unit
C_{IN}	Input pin capacitance	V_{IN} = 0 V, f = 1.0 MHz		15	pF
$C_{I/O}$	I/O pin capacitance	V_{OUT} = 0 V, f = 1.0 MHz		15	pF

Table 15. MAX 7000 5.0-V Device Capacitance: MAX 7000S Devices *Note (12)*

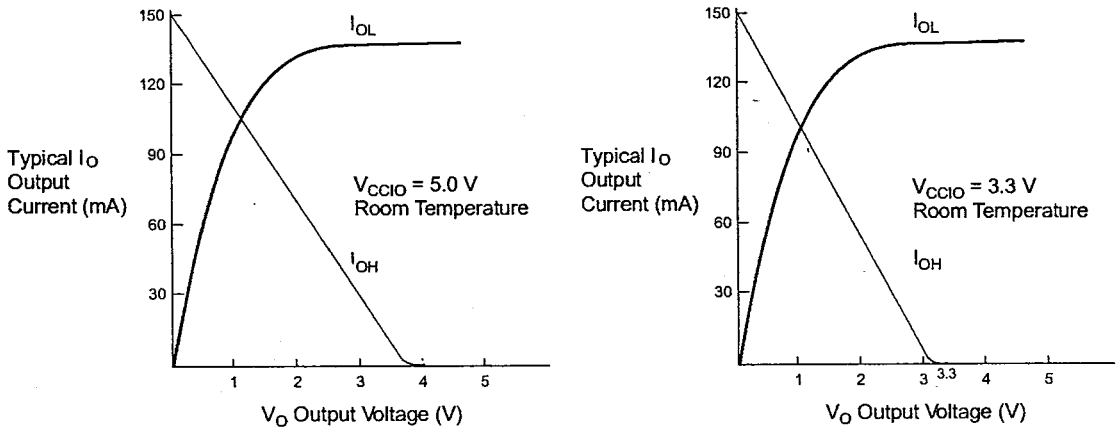
Symbol	Parameter	Conditions	Min	Max	Unit
C_{IN}	Dedicated input pin capacitance	V_{IN} = 0 V, f = 1.0 MHz		10	pF
$C_{I/O}$	I/O pin capacitance	V_{OUT} = 0 V, f = 1.0 MHz		10	pF

Notes to tables:

(1) See the *Operating Requirements for Altera Devices Data Sheet*.
(2) Minimum DC input voltage on I/O pins is –0.5 V and on 4 dedicated input pins is –0.3 V. During transitions, the inputs may undershoot to –2.0 V or overshoot to 7.0 V for input currents less than 100 mA and periods shorter than 20 ns.
(3) Numbers in parentheses are for industrial-temperature-range devices.
(4) V_{CC} must rise monotonically.
(5) 3.3-V I/O operation is not available for 44-pin packages.
(6) The V_{CCISP} parameter applies only to MAX 7000S devices.
(7) During in-system programming, the minimum DC input voltage is –0.3 V.
(8) These values are specified in Table 11 on page 23.
(9) The parameter is measured with 50% of the outputs each sourcing the specified current. The I_{OH} parameter refers to high-level TTL or CMOS output current.
(10) The parameter is measured with 50% of the outputs each sinking the specified current. The I_{OL} parameter refers to low-level TTL or CMOS output current.
(11) When the JTAG interface is enabled in MAX 7000S devices, the input leakage current on the JTAG pins is typically –60 µA.
(12) Capacitance is measured at 25° C and is sample-tested only. The OE1 pin has a maximum capacitance of 20 pF.

Figure 11 shows the typical output drive characteristics of MAX 7000 devices.

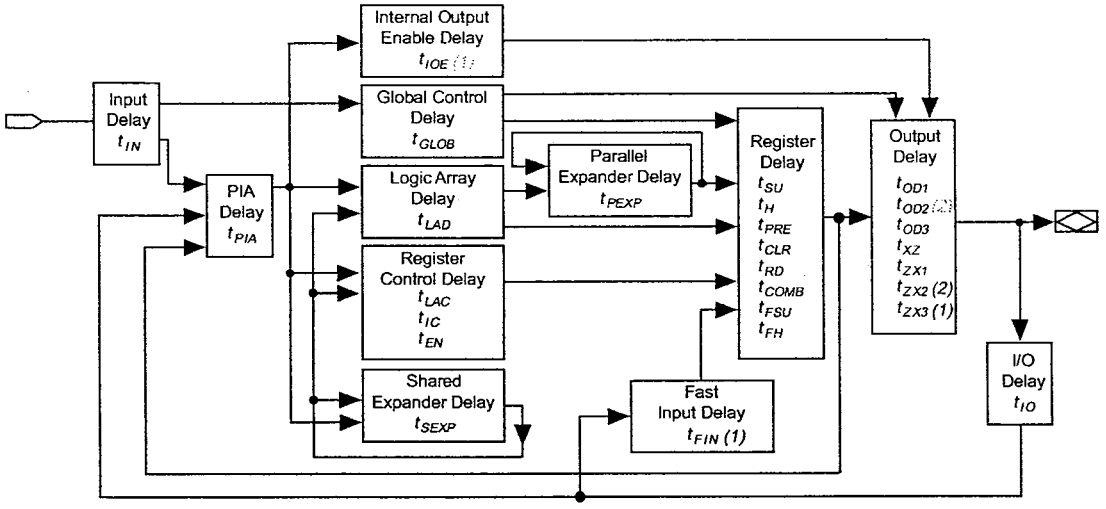
Figure 11. Output Drive Characteristics of 5.0-V MAX 7000 Devices



Timing Model

MAX 7000 device timing can be analyzed with the Quartus or MAX+PLUS II software, with a variety of popular industry-standard EDA simulators and timing analyzers, or with the timing model shown in Figure 12. MAX 7000 devices have fixed internal delays that enable the designer to determine the worst-case timing of any design. The Quartus and MAX+PLUS II software provides timing simulation, point-to-point delay prediction, and detailed timing analysis for a device-wide performance evaluation.

Figure 12. MAX 7000 Timing Model



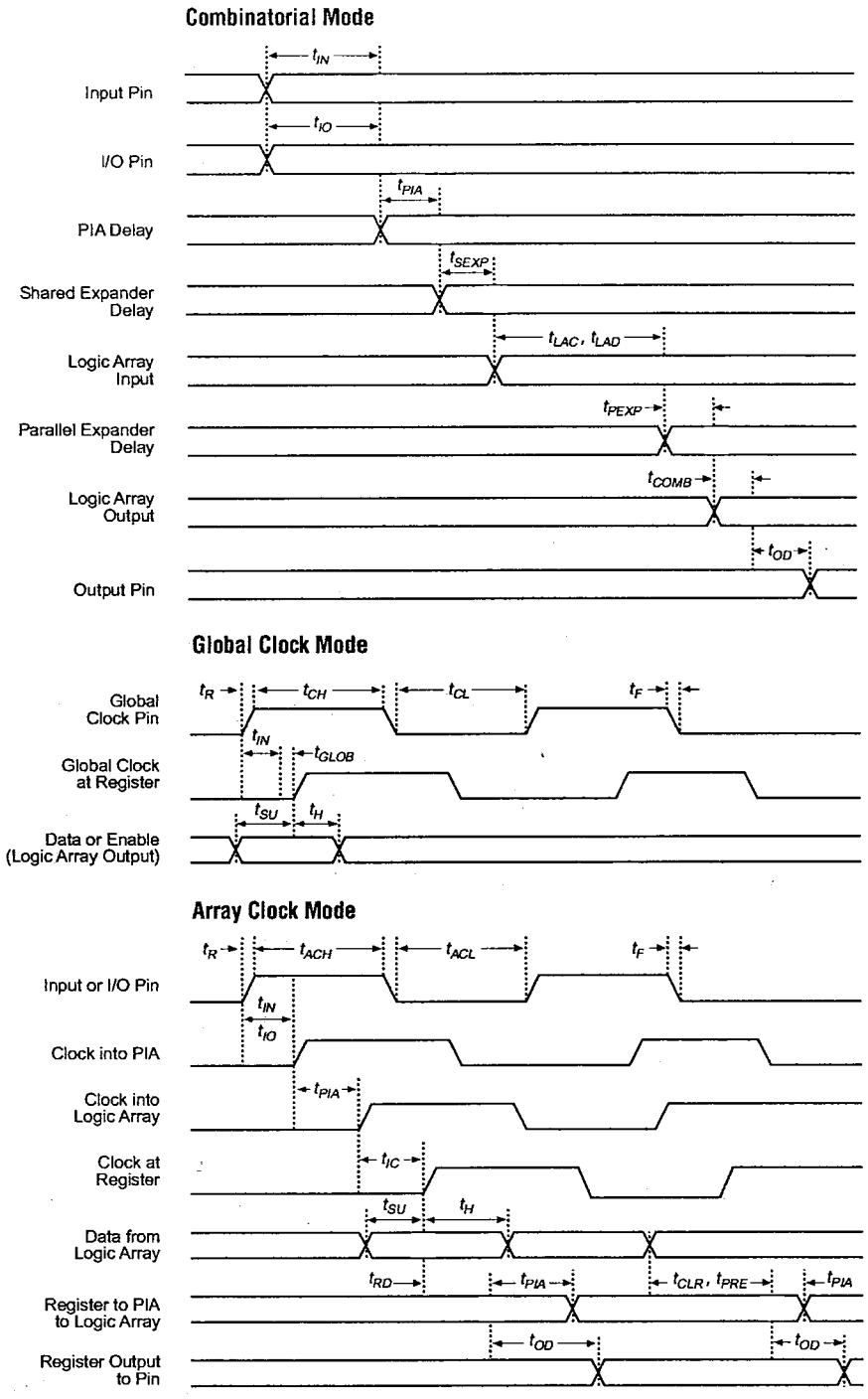
Notes:
(1) Only available in MAX 7000E and MAX 7000S devices.
(2) Not available in 44-pin devices.

The timing characteristics of any signal path can be derived from the timing model and parameters of a particular device. External timing parameters, which represent pin-to-pin timing delays, can be calculated as the sum of internal parameters. Figure 13 shows the internal timing relationship of internal and external delay parameters.

See *Application Note 94 (Understanding MAX 7000 Timing)* for more information.

Figure 13. Switching Waveforms

t_R & t_F < 3 ns. Inputs are driven at 3 V for a logic high and 0 V for a logic low. All timing characteristics are measured at 1.5 V.



Tables 16 through 23 show the MAX 7000 and MAX 7000E AC operating conditions.

Table 16. MAX 7000 & MAX 7000E External Timing Parameters Note (1)

Symbol	Parameter	Conditions	Speed Grade				Unit
			-6		-7		
			Min	Max	Min	Max	
t_{PD1}	Input to non-registered output	C1 = 35 pF		6.0		7.5	ns
t_{PD2}	I/O input to non-registered output	C1 = 35 pF		6.0		7.5	ns
t_{SU}	Global clock setup time		5.0		6.0		ns
t_{H}	Global clock hold time		0.0		0.0		ns
t_{FSU}	Global clock setup time of fast input	*(2)*	2.5		3.0		ns
t_{FH}	Global clock hold time of fast input	*(2)*	0.5		0.5		ns
t_{CO1}	Global clock to output delay	C1 = 35 pF		4.0		4.5	ns
t_{CH}	Global clock high time		2.5		3.0		ns
t_{CL}	Global clock low time		2.5		3.0		ns
t_{ASU}	Array clock setup time		2.5		3.0		ns
t_{AH}	Array clock hold time		2.0		2.0		ns
t_{ACO1}	Array clock to output delay	C1 = 35 pF		6.5		7.5	ns
t_{ACH}	Array clock high time		3.0		3.0		ns
t_{ACL}	Array clock low time		3.0		3.0		ns
t_{CPPW}	Minimum pulse width for clear and preset	*(3)*	3.0		3.0		ns
t_{ODH}	Output data hold time after clock	C1 = 35 pF *(4)*	1.0		1.0		ns
t_{CNT}	Minimum global clock period			6.6		8.0	ns
f_{CNT}	Maximum internal global clock frequency	*(5)*	151.5		125.0		MHz
t_{ACNT}	Minimum array clock period			6.6		8.0	ns
f_{ACNT}	Maximum internal array clock frequency	*(5)*	151.5		125.0		MHz
f_{MAX}	Maximum clock frequency	*(6)*	200		166.7		MHz

Table 17. MAX 7000 & MAX 7000E Internal Timing Parameters

Symbol	Parameter	Conditions	Speed Grade				Unit
			-6		-7		
			Min	Max	Min	Max	
t_{IN}	Input pad and buffer delay			0.4		0.5	ns
t_{IO}	I/O input pad and buffer delay			0.4		0.5	ns
t_{FIN}	Fast input delay	*(2)*		0.8		1.0	ns
t_{SEXP}	Shared expander delay			3.5		4.0	ns
t_{PEXP}	Parallel expander delay			0.8		0.8	ns
t_{LAD}	Logic array delay			2.0		3.0	ns
t_{LAC}	Logic control array delay			2.0		3.0	ns
t_{IOE}	Internal output enable delay	*(2)*				2.0	ns
t_{OD1}	Output buffer and pad delay Slow slew rate = off, V_{CCIO} = 5.0 V	C1 = 35 pF		2.0		2.0	ns
t_{OD2}	Output buffer and pad delay Slow slew rate = off, V_{CCIO} = 3.3 V	C1 = 35 pF *(7)*		2.5		2.5	ns
t_{OD3}	Output buffer and pad delay Slow slew rate = on, V_{CCIO} = 5.0 V or 3.3 V	C1 = 35 pF *(2)*		7.0		7.0	ns
t_{ZX1}	Output buffer enable delay Slow slew rate = off, V_{CCIO} = 5.0 V	C1 = 35 pF		4.0		4.0	ns
t_{ZX2}	Output buffer enable delay Slow slew rate = off, V_{CCIO} = 3.3 V	C1 = 35 pF *(7)*		4.5		4.5	ns
t_{ZX3}	Output buffer enable delay Slow slew rate = on V_{CCIO} = 5.0 V or 3.3 V	C1 = 35 pF *(2)*		9.0		9.0	ns
t_{XZ}	Output buffer disable delay	C1 = 5 pF		4.0		4.0	ns
t_{SU}	Register setup time		3.0		3.0		ns
t_{H}	Register hold time		1.5		2.0		ns
t_{FSU}	Register setup time of fast input	*(2)*	2.5		3.0		ns
t_{FH}	Register hold time of fast input	*(2)*	0.5		0.5		ns
t_{RD}	Register delay			0.8		1.0	ns
t_{COMB}	Combinatorial delay			0.8		1.0	ns
t_{IC}	Array clock delay			2.5		3.0	ns
t_{EN}	Register enable time			2.0		3.0	ns
t_{GLOB}	Global control delay			0.8		1.0	ns
t_{PRE}	Register preset time			2.0		2.0	ns
t_{CLR}	Register clear time			2.0		2.0	ns
t_{PIA}	PIA delay			0.8		1.0	ns
t_{LPA}	Low-power adder	*(8)*		10.0		10.0	ns

Table 18. MAX 7000 & MAX 7000E External Timing Parameters

Symbol	Parameter	Conditions	Speed Grade				Unit
			MAX 7000E (-10P)		MAX 7000 (-10) MAX 7000E (-10)		
			Min	Max	Min	Max	
t_{PD1}	Input to non-registered output	C1 = 35 pF		10.0		10.0	ns
t_{PD2}	I/O input to non-registered output	C1 = 35 pF		10.0		10.0	ns
t_{SU}	Global clock setup time		7.0		8.0		ns
t_{H}	Global clock hold time		0.0		0.0		ns
t_{FSU}	Global clock setup time of fast input	*(2)*	3.0		3.0		ns
t_{FH}	Global clock hold time of fast input	*(2)*	0.5		0.5		ns
t_{CO1}	Global clock to output delay	C1 = 35 pF		5.0		5	ns
t_{CH}	Global clock high time		4.0		4.0		ns
t_{CL}	Global clock low time		4.0		4.0		ns
t_{ASU}	Array clock setup time		2.0		3.0		ns
t_{AH}	Array clock hold time		3.0		3.0		ns
t_{ACO1}	Array clock to output delay	C1 = 35 pF		10.0		10.0	ns
t_{ACH}	Array clock high time		4.0		4.0		ns
t_{ACL}	Array clock low time		4.0		4.0		ns
t_{CPPW}	Minimum pulse width for clear and preset	*(3)*	4.0		4.0		ns
t_{ODH}	Output data hold time after clock	C1 = 35 pF *(4)*	1.0		1.0		ns
t_{CNT}	Minimum global clock period			10.0		10.0	ns
f_{CNT}	Maximum internal global clock frequency	*(5)*	100.0		100.0		MHz
t_{ACNT}	Minimum array clock period			10.0		10.0	ns
f_{ACNT}	Maximum internal array clock frequency	*(5)*	100.0		100.0		MHz
f_{MAX}	Maximum clock frequency	*(6)*	125.0		125.0		MHz

Table 19. MAX 7000 & MAX 7000E Internal Timing Parameters

Symbol	Parameter	Conditions	Speed Grade				Unit
			MAX 7000E (-10P)		MAX 7000 (-10) MAX 7000E (-10)		
			Min	Max	Min	Max	
t_{IN}	Input pad and buffer delay			0.5		1.0	ns
t_{IO}	I/O input pad and buffer delay			0.5		1.0	ns
t_{FIN}	Fast input delay	(2)		1.0		1.0	ns
t_{SEXP}	Shared expander delay			5.0		5.0	ns
t_{PEXP}	Parallel expander delay			0.8		0.8	ns
t_{LAD}	Logic array delay			5.0		5.0	ns
t_{LAC}	Logic control array delay			5.0		5.0	ns
t_{IOE}	Internal output enable delay	(2)		2.0		2.0	ns
t_{OD1}	Output buffer and pad delay Slow slew rate = off V_{CCIO} = 5.0 V	C1 = 35 pF		1.5		2.0	ns
t_{OD2}	Output buffer and pad delay Slow slew rate = off V_{CCIO} = 3.3 V	C1 = 35 pF (7)		2.0		2.5	ns
t_{OD3}	Output buffer and pad delay Slow slew rate = on V_{CCIO} = 5.0 V or 3.3 V	C1 = 35 pF (2)		5.5		6.0	ns
t_{ZX1}	Output buffer enable delay Slow slew rate = off V_{CCIO} = 5.0 V	C1 = 35 pF		5.0		5.0	ns
t_{ZX2}	Output buffer enable delay Slow slew rate = off V_{CCIO} = 3.3 V	C1 = 35 pF (7)		5.5		5.5	ns
t_{ZX3}	Output buffer enable delay Slow slew rate = on V_{CCIO} = 5.0 V or 3.3 V	C1 = 35 pF (2)		9.0		9.0	ns
t_{XZ}	Output buffer disable delay	C1 = 5 pF		5.0		5.0	ns
t_{SU}	Register setup time		2.0		3.0		ns
t_{H}	Register hold time		3.0		3.0		ns
t_{FSU}	Register setup time of fast input	(2)	3.0		3.0		ns
t_{FH}	Register hold time of fast input	(2)	0.5		0.5		ns
t_{RD}	Register delay			2.0		1.0	ns
t_{COMB}	Combinatorial delay			2.0		1.0	ns
t_{IC}	Array clock delay			5.0		5.0	ns
t_{EN}	Register enable time			5.0		5.0	ns
t_{GLOB}	Global control delay			1.0		1.0	ns
t_{PRE}	Register preset time			3.0		3.0	ns
t_{CLR}	Register clear time			3.0		3.0	ns
t_{PIA}	PIA delay			1.0		1.0	ns
t_{LPA}	Low-power adder	(6)		11.0		11.0	ns

Table 20. MAX 7000 & MAX 7000E External Timing Parameters

Symbol	Parameter	Conditions	Speed Grade				Unit
			MAX 7000E (-12P)		MAX 7000 (-12) MAX 7000E (-12)		
			Min	Max	Min	Max	
t_{PD1}	Input to non-registered output	C1 = 35 pF		12.0		12.0	ns
t_{PD2}	I/O input to non-registered output	C1 = 35 pF		12.0		12.0	ns
t_{SU}	Global clock setup time		7.0		10.0		ns
t_{H}	Global clock hold time		0.0		0.0		ns
t_{FSU}	Global clock setup time of fast input	*(2)*	3.0		3.0		ns
t_{FH}	Global clock hold time of fast input	*(2)*	0.0		0.0		ns
t_{CO1}	Global clock to output delay	C1 = 35 pF		6.0		6.0	ns
t_{CH}	Global clock high time		4.0		4.0		ns
t_{CL}	Global clock low time		4.0		4.0		ns
t_{ASU}	Array clock setup time		3.0		4.0		ns
t_{AH}	Array clock hold time		4.0		4.0		ns
t_{ACO1}	Array clock to output delay	C1 = 35 pF		12.0		12.0	ns
t_{ACH}	Array clock high time		5.0		5.0		ns
t_{ACL}	Array clock low time		5.0		5.0		ns
t_{CPPW}	Minimum pulse width for clear and preset	*(3)*	5.0		5.0		ns
t_{ODH}	Output data hold time after clock	C1 = 35 pF *(4)*	1.0		1.0		ns
t_{CNT}	Minimum global clock period			11.0		11.0	ns
f_{CNT}	Maximum internal global clock frequency	*(5)*	90.9		90.9		MHz
t_{ACNT}	Minimum array clock period			11.0		11.0	ns
f_{ACNT}	Maximum internal array clock frequency	*(5)*	90.9		90.9		MHz
f_{MAX}	Maximum clock frequency	*(6)*	125.0		125.0		MHz

Table 21. MAX 7000 & MAX 7000E Internal Timing Parameters

Symbol	Parameter	Conditions	Speed Grade				Unit
			MAX 7000E (-12P)		MAX 7000 (-12) MAX 7000E (-12)		
			Min	Max	Min	Max	
t_{IN}	Input pad and buffer delay			1.0		2.0	ns
t_{IO}	I/O input pad and buffer delay			1.0		2.0	ns
t_{FIN}	Fast input delay	(2)		1.0		1.0	ns
t_{SEXP}	Shared expander delay			7.0		7.0	ns
t_{PEXP}	Parallel expander delay			1.0		1.0	ns
t_{LAD}	Logic array delay			7.0		5.0	ns
t_{LAC}	Logic control array delay			5.0		5.0	ns
t_{IOE}	Internal output enable delay	(2)		2.0		2.0	ns
t_{OD1}	Output buffer and pad delay Slow slew rate = off V_{CCIO} = 5.0 V	C1 = 35 pF		1.0		3.0	ns
t_{OD2}	Output buffer and pad delay Slow slew rate = off V_{CCIO} = 3.3 V	C1 = 35 pF (7)		2.0		4.0	ns
t_{OD3}	Output buffer and pad delay Slow slew rate = on V_{CCIO} = 5.0 V or 3.3 V	C1 = 35 pF (2)		5.0		7.0	ns
t_{ZX1}	Output buffer enable delay Slow slew rate = off V_{CCIO} = 5.0 V	C1 = 35 pF		6.0		6.0	ns
t_{ZX2}	Output buffer enable delay Slow slew rate = off V_{CCIO} = 3.3 V	C1 = 35 pF (7)		7.0		7.0	ns
t_{ZX3}	Output buffer enable delay Slow slew rate = on V_{CCIO} = 5.0 V or 3.3 V	C1 = 35 pF (2)		10.0		10.0	ns
t_{XZ}	Output buffer disable delay	C1 = 5 pF		6.0		6.0	ns
t_{SU}	Register setup time		1.0		4.0		ns
t_{H}	Register hold time		6.0		4.0		ns
t_{FSU}	Register setup time of fast input	(2)	4.0		2.0		ns
t_{FH}	Register hold time of fast input	(2)	0.0		2.0		ns
t_{RD}	Register delay			2.0		1.0	ns
t_{COMB}	Combinatorial delay			2.0		1.0	ns
t_{IC}	Array clock delay			5.0		5.0	ns
t_{EN}	Register enable time			7.0		5.0	ns
t_{GLOB}	Global control delay			2.0		0.0	ns
t_{PRE}	Register preset time			4.0		3.0	ns
t_{CLR}	Register clear time			4.0		3.0	ns
t_{PIA}	PIA delay			1.0		1.0	ns
t_{LPA}	Low-power adder	(8)		12.0		12.0	ns

Table 22. MAX 7000 & MAX 7000E External Timing Parameters

Symbol	Parameter	Conditions	Speed Grade						Unit
			-15		-15T		-20		
			Min	Max	Min	Max	Min	Max	
t_{PD1}	Input to non-registered output	C1 = 35 pF		15.0		15.0		20.0	ns
t_{PD2}	I/O input to non-registered output	C1 = 35 pF		15.0		15.0		20.0	ns
t_{SU}	Global clock setup time		11.0		11.0		12.0		ns
t_{H}	Global clock hold time		0.0		0.0		0.0		ns
t_{FSU}	Global clock setup time of fast input	*(2)*	3.0		–		5.0		ns
t_{FH}	Global clock hold time of fast input	*(2)*	0.0		–		0.0		ns
t_{CO1}	Global clock to output delay	C1 = 35 pF		8.0		8.0		12.0	ns
t_{CH}	Global clock high time		5.0		6.0		6.0		ns
t_{CL}	Global clock low time		5.0		6.0		6.0		ns
t_{ASU}	Array clock setup time		4.0		4.0		5.0		ns
t_{AH}	Array clock hold time		4.0		4.0		5.0		ns
t_{ACO1}	Array clock to output delay	C1 = 35 pF		15.0		15.0		20.0	ns
t_{ACH}	Array clock high time		6.0		6.5		8.0		ns
t_{ACL}	Array clock low time		6.0		6.5		8.0		ns
t_{CPPW}	Minimum pulse width for clear and preset	*(3)*	6.0		6.5		8.0		ns
t_{ODH}	Output data hold time after clock	C1 = 35 pF *(4)*	1.0		1.0		1.0		ns
t_{CNT}	Minimum global clock period			13.0		13.0		16.0	ns
f_{CNT}	Maximum internal global clock frequency	*(5)*	76.9		76.9		62.5		MHz
t_{ACNT}	Minimum array clock period			13.0		13.0		16.0	ns
f_{ACNT}	Maximum internal array clock frequency	*(5)*	76.9		76.9		62.5		MHz
f_{MAX}	Maximum clock frequency	*(6)*	100		83.3		83.3		MHz

Table 23. MAX 7000 & MAX 7000E Internal Timing Parameters

Symbol	Parameter	Conditions	Speed Grade						Unit
			-15		-15T		-20		
			Min	Max	Min	Max	Min	Max	
t_{IN}	Input pad and buffer delay			2.0		2.0		3.0	ns
t_{IO}	I/O input pad and buffer delay			2.0		2.0		3.0	ns
t_{FIN}	Fast input delay	*(2)*		2.0		–		4.0	ns
t_{SEXP}	Shared expander delay			8.0		10.0		9.0	ns
t_{PEXP}	Parallel expander delay			1.0		1.0		2.0	ns
t_{LAD}	Logic array delay			6.0		6.0		8.0	ns
t_{LAC}	Logic control array delay			6.0		6.0		8.0	ns
t_{IOE}	Internal output enable delay	*(2)*		3.0		–		4.0	ns
t_{OD1}	Output buffer and pad delay Slow slew rate = off V_{CCIO} = 5.0 V	C1 = 35 pF		4.0		4.0		5.0	ns
t_{OD2}	Output buffer and pad delay Slow slew rate = off V_{CCIO} = 3.3 V	C1 = 35 pF *(7)*		5.0		–		6.0	ns
t_{OD3}	Output buffer and pad delay Slow slew rate = on V_{CCIO} = 5.0 V or 3.3 V	C1 = 35 pF *(2)*		8.0		–		9.0	ns
t_{ZX1}	Output buffer enable delay Slow slew rate = off V_{CCIO} = 5.0 V	C1 = 35 pF		6.0		6.0		10.0	ns
t_{ZX2}	Output buffer enable delay Slow slew rate = off V_{CCIO} = 3.3 V	C1 = 35 pF *(7)*		7.0		–		11.0	ns
t_{ZX3}	Output buffer enable delay Slow slew rate = on V_{CCIO} = 5.0 V or 3.3 V	C1 = 35 pF *(2)*		10.0		–		14.0	ns
t_{XZ}	Output buffer disable delay	C1 = 5 pF		6.0		6.0		10.0	ns
t_{SU}	Register setup time		4.0		4.0		4.0		ns
t_{H}	Register hold time		4.0		4.0		5.0		ns
t_{FSU}	Register setup time of fast input	*(2)*	2.0		–		4.0		ns
t_{FH}	Register hold time of fast input	*(2)*	2.0		–		3.0		ns
t_{RD}	Register delay			1.0		1.0		1.0	ns
t_{COMB}	Combinatorial delay			1.0		1.0		1.0	ns
t_{IC}	Array clock delay			6.0		6.0		8.0	ns
t_{EN}	Register enable time			6.0		6.0		8.0	ns
t_{GLOB}	Global control delay			1.0		1.0		3.0	ns
t_{PRE}	Register preset time			4.0		4.0		4.0	ns
t_{CLR}	Register clear time			4.0		4.0		4.0	ns
t_{PIA}	PIA delay			2.0		2.0		3.0	ns
t_{LPA}	Low-power adder	*(8)*		13.0		15.0		15.0	ns

Dedicated Pin	84-Pin PLCC	100-Pin PQFP	100-Pin TQFP (1), (2)	160-Pin PQFP
INPUT/GCLK1	83	89	87	139
INPUT/GCLRn	1	91	89	141
INPUT/OE1	84	90	88	140
INPUT/OE2/GCLK2	2	92	90	142
TDI (3)	14	6	4	9
TMS (3)	23	17	15	22
TCK (3)	62	64	62	99
TDO (3)	71	75	73	112
GNDINT	42, 82	40, 88	38, 86	60, 138
GNDIO	7, 19, 32,47, 59, 72	13, 28, 45, 61, 76, 97	11, 26, 43, 59, 74, 95	17, 42, 66, 95, 113, 148
VCCINT (5.0 V only)	3, 43	41, 93	39, 91	61, 143
VCCIO (3.3 V or 5.0 V)	13, 26, 38, 53, 66, 78	5, 20, 36, 53, 68, 84	3, 18, 34, 51, 66, 82	8, 26, 55, 79, 104, 133
No Connect (N.C.)	–	–	–	1, 2, 3, 4, 5, 6, 7, 34, 35, 36, 37, 38, 39, 40, 44, 45, 46, 47, 74, 75, 76, 77, 81, 82, 83, 84, 85, 86, 87, 114, 115, 116, 117, 118, 119, 120, 124, 125, 126, 127, 154, 155, 156, 157
Total User I/O Pins (4)	64	80	80	96

LAB	MC	84-Pin PLCC	100-Pin PQFP	100-Pin TQFP (1), (2)	160-Pin PQFP
A	1	–	4	2	160
A	2	–	–	–	–
A	3	12	3	1	159
A	4	–	–	–	158
A	5	11	2	100	153
A	6	10	1	99	152
A	7	–	–	–	–
A	8	9	100	98	151
A	9	–	99	97	150
A	10	–	–	–	–
A	11	8	98	96	149
A	12	–	–	–	147
A	13	6	96	94	146
A	14	5	95	93	145
A	15	–	–	–	–
A	16	4	94	92	144
B	17	22	16	14	21
B	18	–	–	–	–
B	19	21	15	13	20
B	20	–	–	–	19
B	21	20	14	12	18
B	22	–	12	10	16
B	23	–	–	–	–
B	24	18	11	9	15
B	25	17	10	8	14
B	26	–	–	–	–
B	27	16	9	7	13
B	28	–	–	–	12
B	29	15	8	6	11
B	30	–	7	5	10
B	31	–	–	–	–
B	32	14 (3)	6 (3)	4 (3)	9 (3)
C	33	–	27	25	41
C	34	–	–	–	–
C	35	31	26	24	33
C	36	–	–	–	32
C	37	30	25	23	31
C	38	29	24	22	30
C	39	–	–	–	–
C	40	28	23	21	29
C	41	–	22	20	28
C	42	–	–	–	–
C	43	27	21	19	27
C	44	–	–	–	25
C	45	25	19	17	24
C	46	24	18	16	23
C	47	–	–	–	–
C	48	23 (3)	17 (3)	15 (3)	22 (3)

LAB	MC	84-Pin PLCC	100-Pin PQFP	100-Pin TQFP (1), (2)	160-Pin PQFP
D	49	41	39	37	59
D	50	–	–	–	–
D	51	40	38	36	58
D	52	–	–	–	57
D	53	39	37	35	56
D	54	–	35	33	54
D	55	–	–	–	–
D	56	37	34	32	53
D	57	36	33	31	52
D	58	–	–	–	–
D	59	35	32	30	51
D	60	–	–	–	50
D	61	34	31	29	49
D	62	–	30	28	48
D	63	–	–	–	–
D	64	33	29	27	43
E	65	44	42	40	62
E	66	–	–	–	–
E	67	45	43	41	63
E	68	–	–	–	64
E	69	46	44	42	65
E	70	–	46	44	67
E	71	–	–	–	–
E	72	48	47	45	68
E	73	49	48	46	69
E	74	–	–	–	–
E	75	50	49	47	70
E	76	–	–	–	71
E	77	51	50	48	72
E	78	–	51	49	73
E	79	–	–	–	–
E	80	52	52	50	78
F	81	–	54	52	80
F	82	–	–	–	–
F	83	54	55	53	88
F	84	–	–	–	89
F	85	55	56	54	90
F	86	56	57	55	91
F	87	–	–	–	–
F	88	57	58	56	92
F	89	–	59	57	93
F	90	–	–	–	–
F	91	58	60	58	94
F	92	–	–	–	96
F	93	60	62	60	97
F	94	61	63	61	98
F	95	–	–	–	–
F	96	62 (3)	64 (3)	62 (3)	99 (3)

LAB	MC	84-Pin PLCC	100-Pin PQFP	100-Pin TQFP (1), (2)	160-Pin PQFP
G	97	63	65	63	100
G	98	–	–	–	–
G	99	64	66	64	101
G	100	–	–	–	102
G	101	65	67	65	103
G	102	–	69	67	105
G	103	–	–	–	–
G	104	67	70	68	106
G	105	68	71	69	107
G	106	–	–	–	–
G	107	69	72	70	108
G	108	–	–	–	109
G	109	70	73	71	110
G	110	–	74	72	111
G	111	–	–	–	–
G	112	71 (3)	75 (3)	73 (3)	112 (3)
H	113	–	77	75	121
H	114	–	–	–	–
H	115	73	78	76	122
H	116	–	–	–	123
H	117	74	79	77	128
H	118	75	80	78	129
H	119	–	–	–	–
H	120	76	81	79	130
H	121	–	82	80	131
H	122	–	–	–	–
H	123	77	83	81	132
H	124	–	–	–	134
H	125	79	85	83	135
H	126	80	86	84	136
H	127	–	–	–	–
H	128	81	87	85	137

Notes:
(1) A complete thermal analysis should be performed before committing a design to this device package.
(2) EPM7128E devices are not available in the 100-pin TQFP package.
(3) This JTAG pin applies to MAX 7000S devices only and this pin may function as either a JTAG port or a user I/O pin. If the device is configured to use the JTAG ports for boundary-scan testing or for ISP, this pin is not available as a user I/O pin.
(4) The user I/O pin count includes dedicated input pins and all I/O pins.

Table 28. EPM7128S External Timing Parameters

Symbol	Parameter	Conditions	Speed Grade								Unit
			-6		-7		-10		-15		
			Min	Max	Min	Max	Min	Max	Min	Max	
t_{PD1}	Input to non-registered output	C1 = 35 pF		6.0		7.5		10.0		15.0	ns
t_{PD2}	I/O input to non-registered output	C1 = 35 pF		6.0		7.5		10.0		15.0	ns
t_{SU}	Global clock setup time		3.4		6.0		7.0		11.0		ns
t_{H}	Global clock hold time		0.0		0.0		0.0		0.0		ns
t_{FSU}	Global clock setup time of fast input		2.5		3.0		3.0		3.0		ns
t_{FH}	Global clock hold time of fast input		0.0		0.5		0.5		0.0		ns
t_{CO1}	Global clock to output delay	C1 = 35 pF		4.0		4.5		5.0		8.0	ns
t_{CH}	Global clock high time		3.0		3.0		4.0		5.0		ns
t_{CL}	Global clock low time		3.0		3.0		4.0		5.0		ns
t_{ASU}	Array clock setup time		0.9		3.0		2.0		4.0		ns
t_{AH}	Array clock hold time		1.8		2.0		5.0		4.0		ns
t_{ACO1}	Array clock to output delay	C1 = 35 pF		6.5		7.5		10.0		15.0	ns
t_{ACH}	Array clock high time		3.0		3.0		4.0		6.0		ns
t_{ACL}	Array clock low time		3.0		3.0		4.0		6.0		ns
t_{CPPW}	Minimum pulse width for clear and preset	*(1)*	3.0		3.0		4.0		6.0		ns
t_{ODH}	Output data hold time after clock	C1 = 35 pF *(2)*	1.0		1.0		1.0		1.0		ns
t_{CNT}	Minimum global clock period			6.8		8.0		10.0		13.0	ns
f_{CNT}	Maximum internal global clock frequency	*(3)*	147.1		125.0		100.0		76.9		MHz
t_{ACNT}	Minimum array clock period			6.8		8.0		10.0		13.0	ns
f_{ACNT}	Maximum internal array clock frequency	*(3)*	147.1		125.0		100.0		76.9		MHz
f_{MAX}	Maximum clock frequency	*(4)*	166.7		166.7		125.0		100.0		MHz

Table 29. EPM7128S Internal Timing Parameters

Symbol	Parameter	Conditions	Speed Grade								Unit
			-6		-7		-10		-15		
			Min	Max	Min	Max	Min	Max	Min	Max	
t_{IN}	Input pad and buffer delay			0.2		0.5		0.5		2.0	ns
t_{IO}	I/O input pad and buffer delay			0.2		0.5		0.5		2.0	ns
t_{FIN}	Fast input delay			2.6		1.0		1.0		2.0	ns
t_{SEXP}	Shared expander delay			3.7		4.0		5.0		8.0	ns
t_{PEXP}	Parallel expander delay			1.1		0.8		0.8		1.0	ns
t_{LAD}	Logic array delay			3.0		3.0		5.0		6.0	ns
t_{LAC}	Logic control array delay			3.0		3.0		5.0		6.0	ns
t_{IOE}	Internal output enable delay			0.7		2.0		2.0		3.0	ns
t_{OD1}	Output buffer and pad delay	C1 = 35 pF		0.4		2.0		1.5		4.0	ns
t_{OD2}	Output buffer and pad delay	C1 = 35 pF (5)		0.9		2.5		2.0		5.0	ns
t_{OD3}	Output buffer and pad delay	C1 = 35 pF		5.4		7.0		5.5		8.0	ns
t_{ZX1}	Output buffer enable delay	C1 = 35 pF		4.0		4.0		5.0		6.0	ns
t_{ZX2}	Output buffer enable delay	C1 = 35 pF (5)		4.5		4.5		5.5		7.0	ns
t_{ZX3}	Output buffer enable delay	C1 = 35 pF		9.0		9.0		9.0		10.0	ns
t_{XZ}	Output buffer disable delay	C1 = 5 pF		4.0		4.0		5.0		6.0	ns
t_{SU}	Register setup time		1.0		3.0		2.0		4.0		ns
t_{H}	Register hold time		1.7		2.0		5.0		4.0		ns
t_{FSU}	Register setup time of fast input		1.9		3.0		3.0		2.0		ns
t_{FH}	Register hold time of fast input		0.6		0.5		0.5		1.0		ns
t_{RD}	Register delay			1.4		1.0		2.0		1.0	ns
t_{COMB}	Combinatorial delay			1.0		1.0		2.0		1.0	ns
t_{IC}	Array clock delay			3.1		3.0		5.0		6.0	ns
t_{EN}	Register enable time			3.0		3.0		5.0		6.0	ns
t_{GLOB}	Global control delay			2.0		1.0		1.0		1.0	ns
t_{PRE}	Register preset time			2.4		2.0		3.0		4.0	ns
t_{CLR}	Register clear time			2.4		2.0		3.0		4.0	ns
t_{PIA}	PIA delay	(6)		1.4		1.0		1.0		2.0	ns
t_{LPA}	Low-power adder	(7)		11.0		10.0		11.0		13.0	ns

Notes to tables:

(1) This minimum pulse width for preset and clear applies for both global clear and array controls. The t_{LPA} parameter must be added to this minimum width if the clear or reset signal incorporates the t_{LAD} parameter into the signal path.

(2) This parameter is a guideline that is sample-tested only and is based on extensive device characterization. This parameter applies for both global and array clocking.

(3) Measured with a 16-bit loadable, enabled, up/down counter programmed into each LAB.

(4) The f_{MAX} values represent the highest frequency for pipelined data.

(5) Operating conditions: $V_{CCIO} = 3.3\ V \pm 10\%$ for commercial and industrial use.

(6) For EPM7064S-5, EPM7064S-6, EPM7128S-6, EPM7160S-6, EPM7160S-7, EPM7192S-7, and EPM7256S-7 devices, these values are specified for a PIA fan-out of one LAB (16 macrocells). For each additional LAB fan-out in these devices, add an additional 0.1 ns to the PIA timing value.

(7) The t_{LPA} parameter must be added to the t_{LAD}, t_{LAC}, t_{IC}, t_{EN}, t_{SEXP}, t_{ACL}, and t_{CPPW} parameters for macrocells running in the low-power mode.

Power Consumption

Supply power (P) versus frequency (f_{MAX} in MHz) for MAX 7000 devices is calculated with the following equation:

$$P = P_{INT} + P_{IO} = I_{CCINT} \times V_{CC} + P_{IO}$$

The P_{IO} value, which depends on the device output load characteristics and switching frequency, can be calculated using the guidelines given in *Application Note 74 (Evaluating Power for Altera Devices)*.

The I_{CCINT} value, which depends on the switching frequency and the application logic, is calculated with the following equation:

$$I_{CCINT} =$$

$$A \times MC_{TON} + B \times (MC_{DEV} - MC_{TON}) + C \times MC_{USED} \times f_{MAX} \times tog_{LC}$$

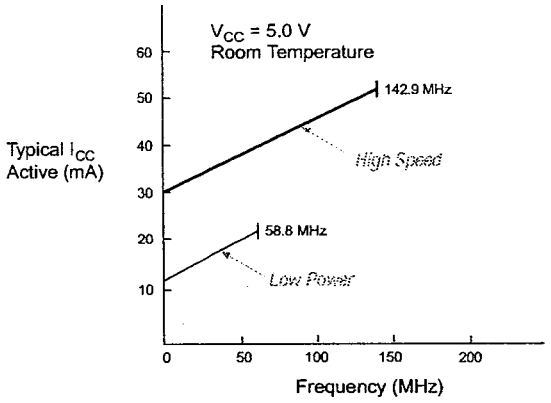
The parameters in this equation are shown below:

MC_{TON}	=	Number of macrocells with the Turbo Bit option turned on, as reported in the MAX+PLUS II Report File (**.rpt**)
MC_{DEV}	=	Number of macrocells in the device
MC_{USED}	=	Total number of macrocells in the design, as reported in the MAX+PLUS II Report File (**.rpt**)
f_{MAX}	=	Highest clock frequency to the device
tog_{LC}	=	Average ratio of logic cells toggling at each clock (typically 0.125)
A, B, C	=	Constants, shown in Table 36

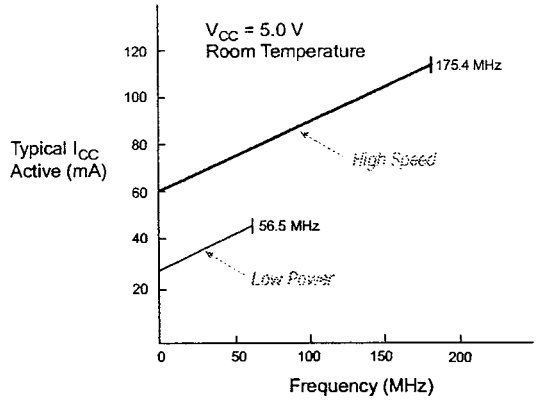
Figure 15 shows typical supply current versus frequency for MAX 7000S devices.

Figure 15. I_{CC} vs. Frequency for MAX 7000S Devices (Part 1 of 2)

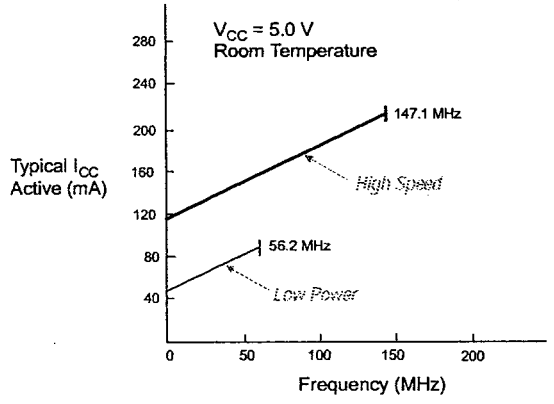
EPM7032S



EPM7064S



EPM7128S



EPM7160S

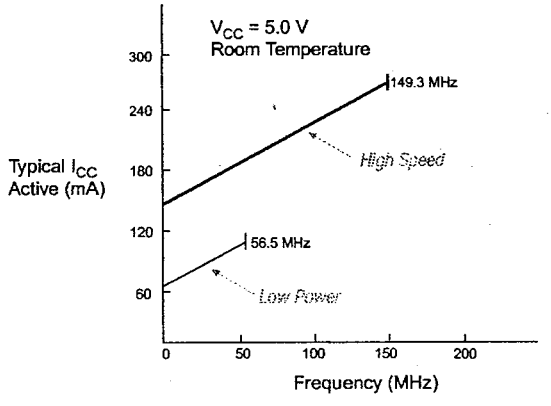


Table 44. EPM7128E & EPM7128S Dedicated Pin-Outs

Dedicated Pin	84-Pin PLCC	100-Pin PQFP	100-Pin TQFP (1), (2)	160-Pin PQFP
INPUT/GCLK1	83	89	87	139
INPUT/GCLRn	1	91	89	141
INPUT/OE1	84	90	88	140
INPUT/OE2/GCLK2	2	92	90	142
TDI (3)	14	6	4	9
TMS (3)	23	17	15	22
TCK (3)	62	64	62	99
TDO (3)	71	75	73	112
GNDINT	42, 82	40, 88	38, 86	60, 138
GNDIO	7, 19, 32,47, 59, 72	13, 28, 45, 61, 76, 97	11, 26, 43, 59, 74, 95	17, 42, 66, 95, 113, 148
VCCINT (5.0 V only)	3, 43	41, 93	39, 91	61, 143
VCCIO (3.3 V or 5.0 V)	13, 26, 38, 53, 66, 78	5, 20, 36, 53, 68, 84	3, 18, 34, 51, 66, 82	8, 26, 55, 79, 104, 133
No Connect (N.C.)	–	–	–	1, 2, 3, 4, 5, 6, 7, 34, 35, 36, 37, 38, 39, 40, 44, 45, 46, 47, 74, 75, 76, 77, 81, 82, 83, 84, 85, 86, 87, 114, 115, 116, 117, 118, 119, 120, 124, 125, 126, 127, 154, 155, 156, 157
Total User I/O Pins (4)	64	80	80	96

Table 45. EPM7128E & EPM7128S I/O Pin-Outs (Part 1 of 2)

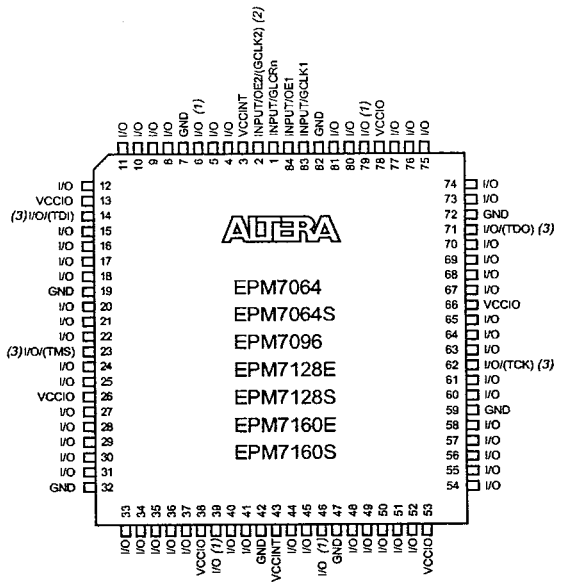
LAB	MC	84-Pin PLCC	100-Pin PQFP	100-Pin TQFP *(1), (2)*	160-Pin PQFP	LAB	MC	84-Pin PLCC	100-Pin PQFP	100-Pin TQFP *(1), (2)*	160-Pin PQFP
A	1	–	4	2	160	C	33	–	27	25	41
	2	–	–	–	–		34	–	–	–	–
	3	12	3	1	159		35	31	26	24	33
	4	–	–	–	158		36	–	–	–	32
	5	11	2	100	153		37	30	25	23	31
	6	10	1	99	152		38	29	24	22	30
	7	–	–	–	–		39	–	–	–	–
	8	9	100	98	151		40	28	23	21	29
	9	–	99	97	150		41	–	22	20	28
	10	–	–	–	–		42	–	–	–	–
	11	8	98	96	149		43	27	21	19	27
	12	–	–	–	147		44	–	–	–	25
	13	6	96	94	146		45	25	19	17	24
	14	5	95	93	145		46	24	18	16	23
	15	–	–	–	–		47	–	–	–	–
	16	4	94	92	144		48	23 *(3)*	17 *(3)*	15 *(3)*	22 *(3)*
B	17	22	16	14	21	D	49	41	39	37	59
	18	–	–	–	–		50	–	–	–	–
	19	21	15	13	20		51	40	38	36	58
	20	–	–	–	19		52	–	–	–	57
	21	20	14	12	18		53	39	37	35	56
	22	–	12	10	16		54	–	35	33	54
	23	–	–	–	–		55	–	–	–	–
	24	18	11	9	15		56	37	34	32	53
	25	17	10	8	14		57	36	33	31	52
	26	–	–	–	–		58	–	–	–	–
	27	16	9	7	13		59	35	32	30	51
	28	–	–	–	12		60	–	–	–	50
	29	15	8	6	11		61	34	31	29	49
	30	–	7	5	10		62	–	30	28	48
	31	–	–	–	–		63	–	–	–	–
	32	14 *(3)*	6 *(3)*	4 *(3)*	9 *(3)*		64	33	29	27	43

Table 45. EPM7128E & EPM7128S I/O Pin-Outs (Part 2 of 2)

LAB	MC	84-Pin PLCC	100-Pin PQFP	100-Pin TQFP (1), (2)	160-Pin PQFP	LAB	MC	84-Pin PLCC	100-Pin PQFP	100-Pin TQFP (1), (2)	160-Pin PQFP
E	65	44	42	40	62	G	97	63	65	63	100
	66	–	–	–	–		98	–	–	–	–
	67	45	43	41	63		99	64	66	64	101
	68	–	–	–	64		100	–	–	–	102
	69	46	44	42	65		101	65	67	65	103
	70	–	46	44	67		102	–	69	67	105
	71	–	–	–	–		103	–	–	–	–
	72	48	47	45	68		104	67	70	68	106
	73	49	48	46	69		105	68	71	69	107
	74	–	–	–	–		106	–	–	–	–
	75	50	49	47	70		107	69	72	70	108
	76	–	–	–	71		108	–	–	–	109
	77	51	50	48	72		109	70	73	71	110
	78	–	51	49	73		110	–	74	72	111
	79	–	–	–	–		111	–	–	–	–
	80	52	52	50	78		112	71 (3)	75 (3)	73 (3)	112 (3)
F	81	–	54	52	80	H	113	–	77	75	121
	82	–	–	–	–		114	–	–	–	–
	83	54	55	53	88		115	73	78	76	122
	84	–	–	–	89		116	–	–	–	123
	85	55	56	54	90		117	74	79	77	128
	86	56	57	55	91		118	75	80	78	129
	87	–	–	–	–		119	–	–	–	–
	88	57	58	56	92		120	76	81	79	130
	89	–	59	57	93		121	–	82	80	131
	90	–	–	–	–		122	–	–	–	–
	91	58	60	58	94		123	77	83	81	132
	92	–	–	–	96		124	–	–	–	134
	93	60	62	60	97		125	79	85	83	135
	94	61	63	61	98		126	80	86	84	136
	95	–	–	–	–		127	–	–	–	–
	96	62 (3)	64 (3)	62 (3)	99 (3)		128	81	87	85	137

Figure 18. 84-Pin Package Pin-Out Diagram

Package outline not drawn to scale. Pin functions in parentheses are for MAX 7000S or MAX 7000E devices only.



84-Pin PLCC

Notes:
(1) Pins 6, 39, 46, and 79 are no-connect (N.C.) pins on EPM7096, EPM7160E, and EPM7160S devices.
(2) This pin is available in MAX 7000E and MAX 7000S devices only.
(3) JTAG ports are available in MAX 7000S devices only.

- Low Supply Voltage Range 1.8 V to 3.6 V
- Ultralow-Power Consumption
  - Active Mode: 160 μA at 1 MHz, 2.2 V
  - Standby Mode: 0.7 μA
  - Off Mode (RAM Retention): 0.1 μA
- Wake-Up From Standby Mode in less than 6 μs
- 16-Bit RISC Architecture, 125 ns Instruction Cycle Time
- Basic Clock Module Configurations:
  - Various Internal Resistors
  - Single External Resistor
  - 32-kHz Crystal
  - High-Frequency Crystal
  - Resonator
  - External Clock Source
- 16-Bit Timer_A With Three Capture/Compare Registers
- On-Chip Comparator for Analog Signal Compare Function or Slope A/D Conversion
- Serial Onboard Programming, No External Programming Voltage Needed Programmable Code Protection by Security Fuse
- Family Members Include:
  MSP430C1101: 1KB ROM, 128B RAM
  MSP430C1111: 2KB ROM, 128B RAM
  MSP430C1121: 4KB ROM, 256B RAM
  MSP430F1101A: 1KB + 128B Flash Memory 128B RAM
  MSP430F1111A: 2KB + 256B Flash Memory 128B RAM
  MSP430F1121A: 4KB + 256B Flash Memory 256B RAM
- Available in a 20-Pin Plastic Small-Outline Wide Body (SOWB) Package, 20-Pin Plastic Small-Outline Thin Package, 20-Pin TVSOP (F11x1A only) and 24-Pin QFN
- For Complete Module Descriptions, Refer to the *MSP430x1xx Family User's Guide*, Literature Number SLAU049

description

The Texas Instruments MSP430 family of ultralow power microcontrollers consist of several devices featuring different sets of peripherals targeted for various applications. The architecture, combined with five low power modes is optimized to achieve extended battery life in portable measurement applications. The device features a powerful 16-bit RISC CPU, 16-bit registers, and constant generators that attribute to maximum code efficiency. The digitally controlled oscillator (DCO) allows wake-up from low-power modes to active mode in less than 6μs.

The MSP430x11x1(A) series is an ultralow-power mixed signal microcontroller with a built-in 16-bit timer, versatile analog comparator and fourteen I/O pins.

Typical applications include sensor systems that capture analog signals, convert them to digital values, and then process the data for display or for transmission to a host system. Stand alone RF sensor front end is another area of application. The I/O port inputs provide single slope A/D conversion capability on resistive sensors.

AVAILABLE OPTIONS

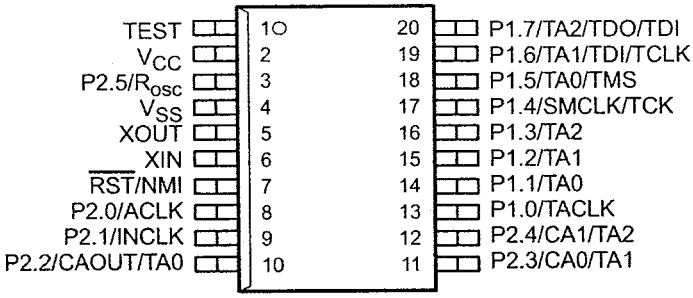
T_A	PACKAGED DEVICES			
	PLASTIC 20-PIN SOWB (DW)	PLASTIC 20-PIN TSSOP (PW)	PLASTIC 20-PIN TVSOP (DGV)	PLASTIC 24-PIN QFN (RGE)
−40°C to 85°C	MSP430C1101IDW MSP430C1111IDW MSP430C1121IDW MSP430F1101AIDW MSP430F1111AIDW MSP430F1121AIDW	MSP430C1101IPW MSP430C1111IPW MSP430C1121IPW MSP430F1101AIPW MSP430F1111AIPW MSP430F1121AIPW	MSP430F1101AIDGV MSP430F1111AIDGV MSP430F1121AIDGV	MSP430C1101IRGE MSP430C1111IRGE MSP430C1121IRGE MSP430F1101AIRGE MSP430F1111AIRGE MSP430F1121AIRGE

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

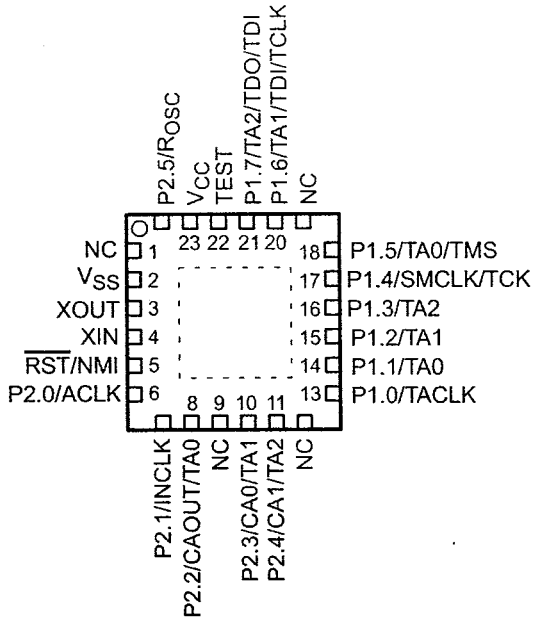
PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 1999 – 2004 Texas Instruments Incorporated

DW, PW, or DGV PACKAGE
(TOP VIEW)

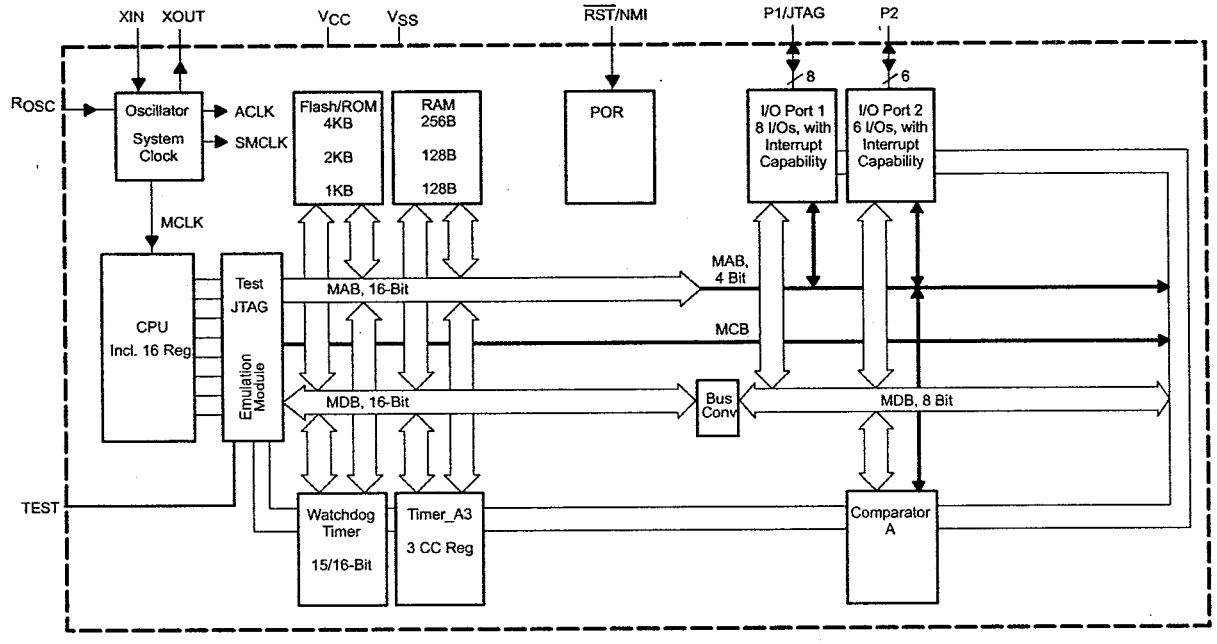


RGE PACKAGE
(TOP VIEW)



Note: NC pins not internally connected
Power Pad connection to V_{SS} recommended

functional block diagram



TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Terminal Functions

TERMINAL				DESCRIPTION
NAME	DW, PW, or DGV NO.	RGE NO.	I/O	
P1.0/TACLK	13	13	I/O	General-purpose digital I/O pin/Timer_A, clock signal TACLK input
P1.1/TA0	14	14	I/O	General-purpose digital I/O pin/Timer_A, capture: CCI0A input, compare: Out0 output/BSL transmit
P1.2/TA1	15	15	I/O	General-purpose digital I/O pin/Timer_A, capture: CCI1A input, compare: Out1 output
P1.3/TA2	16	16	I/O	General-purpose digital I/O pin/Timer_A, capture: CCI2A input, compare: Out2 output
P1.4/SMCLK/TCK	17	17	I/O	General-purpose digital I/O pin/SMCLK signal output/test clock, input terminal for device programming and test
P1.5/TA0/TMS	18	18	I/O	General-purpose digital I/O pin/Timer_A, compare: Out0 output/test mode select, input terminal for device programming and test
P1.6/TA1/TDI/TCLK	19	20	I/O	General-purpose digital I/O pin/Timer_A, compare: Out1 output/test data input or test clock input
P1.7/TA2/TDO/TDI†	20	21	I/O	General-purpose digital I/O pin/Timer_A, compare: Out2 output/test data output terminal or data input during programming
P2.0/ACLK	8	6	I/O	General-purpose digital I/O pin/ACLK output
P2.1/INCLK	9	7	I/O	General-purpose digital I/O pin/Timer_A, clock signal at INCLK
P2.2/CAOUT/TA0	10	8	I/O	General-purpose digital I/O pin/Timer_A, capture: CCI0B input/ comparator_A, output/BSL receive
P2.3/CA0/TA1	11	10	I/O	General-purpose digital I/O pin/Timer_A, compare: Out1 output/ comparator_A, input
P2.4/CA1/TA2	12	11	I/O	General-purpose digital I/O pin/Timer_A, compare: Out2 output/ comparator_A, input
P2.5/R_{OSC}	3	24	I/O	General-purpose digital I/O pin/input for external resistor that defines the DCO nominal frequency
$\overline{RST}$/NMI	7	5	I	Reset or nonmaskable interrupt input
TEST	1	22	I	Selects test mode for JTAG pins on Port1. The device protection fuse is connected to TEST.
V_{CC}	2	23		Supply voltage
V_{SS}	4	2		Ground reference
XIN	6	4	I	Input terminal of crystal oscillator
XOUT	5	3	O	Output terminal of crystal oscillator
QFN Pad	NA	Package Pad	NA	QFN package pad connection to V_{SS} recommended.

† TDO or TDI is selected via JTAG instruction.

short-form description

CPU

The MSP430 CPU has a 16-bit RISC architecture that is highly transparent to the application. All operations, other than program-flow instructions, are performed as register operations in conjunction with seven addressing modes for source operand and four addressing modes for destination operand.

The CPU is integrated with 16 registers that provide reduced instruction execution time. The register-to-register operation execution time is one cycle of the CPU clock.

Four of the registers, R0 to R3, are dedicated as program counter, stack pointer, status register, and constant generator respectively. The remaining registers are general-purpose registers.

Peripherals are connected to the CPU using data, address, and control buses, and can be handled with all instructions.

instruction set

The instruction set consists of 51 instructions with three formats and seven address modes. Each instruction can operate on word and byte data. Table 1 shows examples of the three types of instruction formats; the address modes are listed in Table 2.

Table 1. Instruction Word Formats

Dual operands, source-destination	e.g. ADD R4,R5	R4 + R5 ---> R5
Single operands, destination only	e.g. CALL R8	PC -->(TOS), R8--> PC
Relative jump, un/conditional	e.g. JNE	Jump-on-equal bit = 0

Table 2. Address Mode Descriptions

ADDRESS MODE	S	D	SYNTAX	EXAMPLE	OPERATION
Register	●	●	MOV Rs,Rd	MOV R10,R11	R10 --> R11
Indexed	●	●	MOV X(Rn),Y(Rm)	MOV 2(R5),6(R6)	M(2+R5)--> M(6+R6)
Symbolic (PC relative)	●	●	MOV EDE,TONI		M(EDE) --> M(TONI)
Absolute	●	●	MOV &MEM,&TCDAT		M(MEM) --> M(TCDAT)
Indirect	●		MOV @Rn,Y(Rm)	MOV @R10,Tab(R6)	M(R10) --> M(Tab+R6)
Indirect autoincrement	●		MOV @Rn+,Rm	MOV @R10+,R11	M(R10) --> R11 R10 + 2--> R10
Immediate	●		MOV #X,TONI	MOV #45,TONI	#45 --> M(TONI)

NOTE: S = source D = destination

operating modes

The MSP430 has one active mode and five software selectable low-power modes of operation. An interrupt event can wake up the device from any of the five low-power modes, service the request and restore back to the low-power mode on return from the interrupt program.

The following six operating modes can be configured by software:

- Active mode AM;
  - All clocks are active
- Low-power mode 0 (LPM0);
  - CPU is disabled
    ACLK and SMCLK remain active. MCLK is disabled
- Low-power mode 1 (LPM1);
  - CPU is disabled
    ACLK and SMCLK remain active. MCLK is disabled
    DCO's dc-generator is disabled if DCO not used in active mode
- Low-power mode 2 (LPM2);
  - CPU is disabled
    MCLK and SMCLK are disabled
    DCO's dc-generator remains enabled
    ACLK remains active
- Low-power mode 3 (LPM3);
  - CPU is disabled
    MCLK and SMCLK are disabled
    DCO's dc-generator is disabled
    ACLK remains active
- Low-power mode 4 (LPM4);
  - CPU is disabled
    ACLK is disabled
    MCLK and SMCLK are disabled
    DCO's dc-generator is disabled
    Crystal oscillator is stopped

interrupt vector addresses

The interrupt vectors and the power-up starting address are located in the address range of 0FFFFh–0FFE0h. The vector contains the 16-bit address of the appropriate interrupt handler instruction sequence.

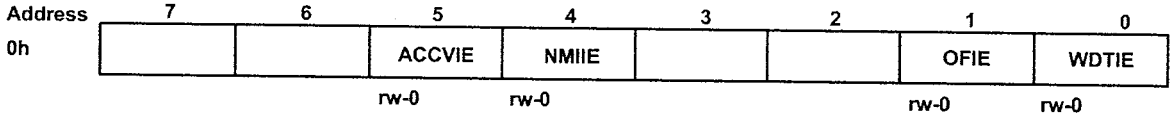
INTERRUPT SOURCE	INTERRUPT FLAG	SYSTEM INTERRUPT	WORD ADDRESS	PRIORITY
Power-up External reset Watchdog Flash Memory	WDTIFG KEYV (see Note 1)	Reset	0FFFEh	15, highest
NMI Oscillator fault Flash memory access violation	NMIIFG OFIFG ACCVIFG (see Notes 1 & 4)	(non)-maskable, (non)-maskable, (non)-maskable	0FFFCh	14
			0FFFAh	13
			0FFF8h	12
Comparator_A	CAIFG	maskable	0FFF6h	11
Watchdog Timer	WDTIFG	maskable	0FFF4h	10
Timer_A3	TACCR0 CCIFG (see Note 2)	maskable	0FFF2h	9
Timer_A3	TACCR1 CCIFG. TACCR2 CCIFG TAIFG (see Notes 1 & 2)	maskable	0FFF0h	8
			0FFEEh	7
			0FFECh	6
			0FFEAh	5
			0FFE8h	4
I/O Port P2 (eight flags; see Note 3)	P2IFG.0 to P2IFG.7 (see Notes 1 & 2)	maskable	0FFE6h	3
I/O Port P1 (eight flags)	P1IFG.0 to P1IFG.7 (see Notes 1 & 2)	maskable	0FFE4h	2
			0FFE2h	1
			0FFE0h	0, lowest

NOTES: 1. Multiple source flags
2. Interrupt flags are located in the module
3. There are eight Port P2 interrupt flags, but only six Port P2 I/O pins (P2.0–5) implemented on the 'C11x1 and 'F11x1A devices.
4. (non)-maskable: the individual interrupt-enable bit can disable an interrupt event, but the general interrupt enable cannot. Nonmaskable: neither the individual nor the general interrupt-enable bit will disable an interrupt event.

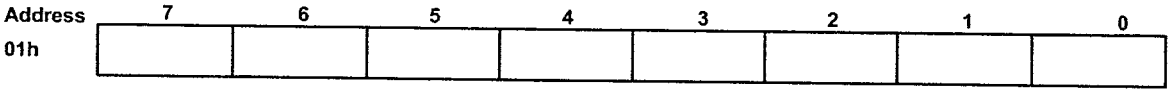
special function registers

Most interrupt and module enable bits are collected into the lowest address space. Special function register bits not allocated to a functional purpose are not physically present in the device. Simple software access is provided with this arrangement.

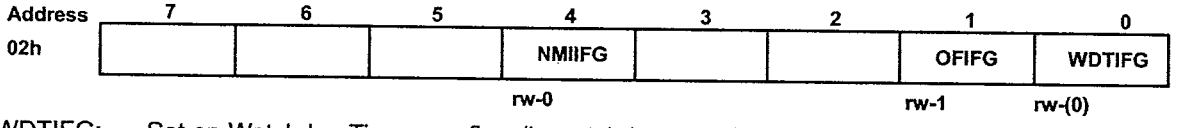
interrupt enable 1 and 2



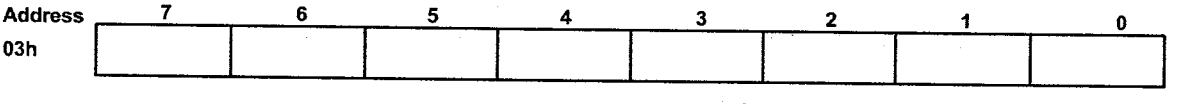
WDTIE: Watchdog Timer interrupt enable. Inactive if watchdog mode is selected. Active if Watchdog Timer is configured in interval timer mode.
OFIE: Oscillator fault enable
NMIIE: (Non)maskable interrupt enable
ACCVIE: Flash access violation interrupt enable



interrupt flag register 1 and 2



WDTIFG: Set on Watchdog Timer overflow (in watchdog mode) or security key violation. Reset on V_{CC} power-up or a reset condition at $\overline{RST}$/NMI pin in reset mode.
OFIFG: Flag set on oscillator fault
NMIIFG: Set via $\overline{RST}$/NMI-pin



Legend		
	rw:	Bit can be read and written.
	rw-0,1:	Bit can be read and written. It is Reset or Set by PUC.
	rw-(0,1):	Bit can be read and written. It is Reset or Set by POR.
		SFR bit is not present in device

memory organization

		MSP430C1101	MSP430C1111	MSP430C1121
Memory Main: interrupt vector Main: code memory	Size ROM ROM	1KB ROM 0FFFFh–0FFE0h 0FFFFh–0FC00h	2KB ROM 0FFFFh–0FFE0h 0FFFFh–0F800h	4KB ROM 0FFFFh–0FFE0h 0FFFFh–0F000h
Information memory	Size Flash	Not applicable	Not applicable	Not applicable
Boot memory	Size ROM	Not applicable	Not applicable	Not applicable
RAM	Size	128 Byte 027Fh – 0200h	128 Byte 027Fh – 0200h	256 Byte 02FFh – 0200h
Peripherals	16-bit 8-bit 8-bit SFR	01FFh – 0100h 0FFh – 010h 0Fh – 00h	01FFh – 0100h 0FFh – 010h 0Fh – 00h	01FFh – 0100h 0FFh – 010h 0Fh – 00h

		MSP430F1101A	MSP430F1111A	MSP430F1121A
Memory Main: interrupt vector Main: code memory	Size Flash Flash	1KB Flash 0FFFFh–0FFE0h 0FFFFh–0FC00h	2KB Flash 0FFFFh–0FFE0h 0FFFFh–0F800h	4KB Flash 0FFFFh–0FFE0h 0FFFFh–0F000h
Information memory	Size Flash	128 Byte 010FFh – 01080h	256 Byte 010FFh – 01000h	256 Byte 010FFh – 01000h
Boot memory	Size ROM	1KB 0FFFh – 0C00h	1KB 0FFFh – 0C00h	1KB 0FFFh – 0C00h
RAM	Size	128 Byte 027Fh – 0200h	128 Byte 027Fh – 0200h	256 Byte 02FFh – 0200h
Peripherals	16-bit 8-bit 8-bit SFR	01FFh – 0100h 0FFh – 010h 0Fh – 00h	01FFh – 0100h 0FFh – 010h 0Fh – 00h	01FFh – 0100h 0FFh – 010h 0Fh – 00h

bootstrap loader (BSL)

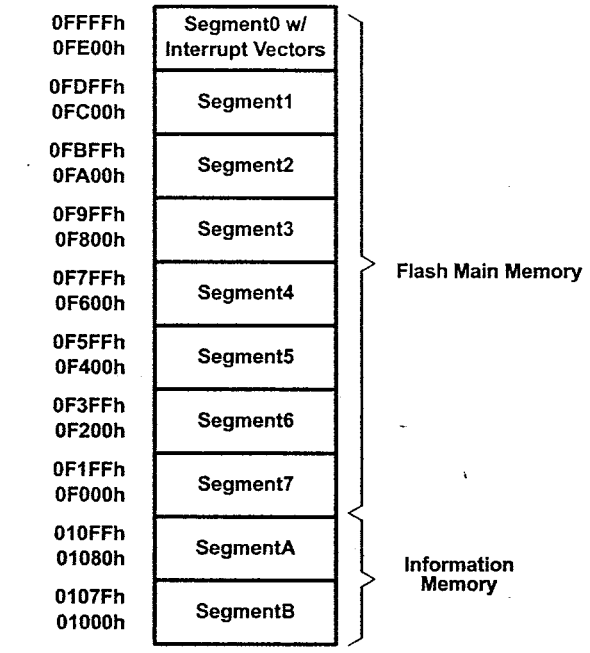
The MSP430 bootstrap loader (BSL) enables users to program the flash memory or RAM using a UART serial interface. Access to the MSP430 memory via the BSL is protected by user-defined password. For complete description of the features of the BSL and its implementation, see the Application report *Features of the MSP430 Bootstrap Loader*, Literature Number SLAA089.

BSL Function	DW, PW & DGV Package Pins	RGE Package Pins
Data Transmit	14 - P1.1	14 - P1.1
Data Receive	10 - P2.2	8 - P2.2

flash memory

The flash memory can be programmed via the JTAG port, the bootstrap loader, or in-system by the CPU. The CPU can perform single-byte and single-word writes to the flash memory. Features of the flash memory include:

- Flash memory has n segments of main memory and two segments of information memory (A and B) of 128 bytes each. Each segment in main memory is 512 bytes in size.
- Segments 0 to n may be erased in one step, or each segment may be individually erased.
- Segments A and B can be erased individually, or as a group with segments 0–n.
  Segments A and B are also called *information memory.*
- New devices may have some bytes programmed in the information memory (needed for test during manufacturing). The user should perform an erase of the information memory prior to the first use.



NOTE: All segments not implemented on all devices.

peripherals

Peripherals are connected to the CPU through data, address, and control busses and can be handled using all instructions. For complete module descriptions, refer to the *MSP430x1xx Family User's Guide*, literature number SLAU049.

oscillator and system clock

The clock system is supported by the basic clock module that includes support for a 32768-Hz watch crystal oscillator, an internal digitally-controlled oscillator (DCO) and a high frequency crystal oscillator. The basic clock module is designed to meet the requirements of both low system cost and low-power consumption. The internal DCO provides a fast turn-on clock source and stabilizes in less than 6 µs. The basic clock module provides the following clock signals:

- Auxiliary clock (ACLK), sourced from a 32768-Hz watch crystal or a high frequency crystal.
- Main clock (MCLK), the system clock used by the CPU.
- Sub-Main clock (SMCLK), the sub-system clock used by the peripheral modules.

digital I/O

There are two 8-bit I/O ports implemented—ports P1 and P2 (only six P2 I/O signals are available on external pins):

- All individual I/O bits are independently programmable.
- Any combination of input, output, and interrupt conditions is possible.
- Edge-selectable interrupt input capability for all the eight bits of port P1 and six bits of port P2.
- Read/write access to port-control registers is supported by all instructions.

NOTE:
Six bits of port P2, P2.0 to P2.5, are available on external pins – but all control and data bits for port P2 are implemented.

watchdog timer

The primary function of the watchdog timer (WDT) module is to perform a controlled system restart after a software problem occurs. If the selected time interval expires, a system reset is generated. If the watchdog function is not needed in an application, the module can be configured as an interval timer and can generate interrupts at selected time intervals.

comparator_A

The primary function of the comparator_A module is to support precision slope analog-to-digital conversions, battery-voltage supervision, and monitoring of external analog signals.

timer_A3

Timer_A3 is a 16-bit timer/counter with three capture/compare registers. Timer_A3 can support multiple capture/compares, PWM outputs, and interval timing. Timer_A3 also has extensive interrupt capabilities. Interrupts may be generated from the counter on overflow conditions and from each of the capture/compare registers.

Timer_A3 Signal Connections							
Input Pin Number		Device Input Signal	Module Input Name	Module Block	Module Output Signal	Output Pin Number	
DW, PW, DGV	RGE					DW, PW DGV	RGE
13 - P1.0	13 - P1.0	TACLK	TACLK	Timer	NA		
		ACLK	ACLK				
		SMCLK	SMCLK				
9 - P2.1	7 - P2.1	INCLK	INCLK				
14 - P1.1	14 - P1.1	TA0	CCI0A	CCR0	TA0	14 - P1.1	14 - P1.1
10 - P2.2	8 - P2.2	TA0	CCI0B			18 - P1.5	18 - P1.5
		V_{SS}	GND				
		V_{CC}	V_{CC}				
15 - P1.2	15 - P1.2	TA1	CCI1A	CCR1	TA1	11 - P2.3	10 - P2.3
		CAOUT (internal)	CCI1B			15 - P1.2	15 - P1.2
		V_{SS}	GND			19 - P1.6	20 - P1.6
		V_{CC}	V_{CC}				
16 - P1.3	16 - P1.3	TA2	CCI2A	CCR2	TA2	12 - P2.4	11 - P2.4
		ACLK (internal)	CCI2B			16 - P1.3	16 - P1.3
		V_{SS}	GND			20 - P1.7	21 - P1.7
		V_{CC}	V_{CC}				

peripheral file map

PERIPHERALS WITH WORD ACCESS			
Timer_A	Reserved		017Eh
	Reserved		017Ch
	Reserved		017Ah
	Reserved		0178h
	Capture/compare register	TACCR2	0176h
	Capture/compare register	TACCR1	0174h
	Capture/compare register	TACCR0	0172h
	Timer_A register	TAR	0170h
	Reserved		016Eh
	Reserved		016Ch
	Reserved		016Ah
	Reserved		0168h
	Capture/compare control	TACCTL2	0166h
	Capture/compare control	TACCTL1	0164h
	Capture/compare control	TACCTL0	0162h
	Timer_A control	TACTL	0160h
	Timer_A interrupt vector	TAIV	012Eh
Flash Memory	Flash control 3	FCTL3	012Ch
	Flash control 2	FCTL2	012Ah
	Flash control 1	FCTL1	0128h
Watchdog	Watchdog/timer control	WDTCTL	0120h
PERIPHERALS WITH BYTE ACCESS			
Comparator_A	Comparator_A port disable	CAPD	05Bh
	Comparator_A control 2	CACTL2	05Ah
	Comparator_A control 1	CACTL1	059h
Basic Clock	Basic clock system control 2	BCSCTL2	058h
	Basic clock system control 1	BCSCTL1	057h
	DCO clock frequency control	DCOCTL	056h
Port P2	Port P2 selection	P2SEL	02Eh
	Port P2 interrupt enable	P2IE	02Dh
	Port P2 interrupt edge select	P2IES	02Ch
	Port P2 interrupt flag	P2IFG	02Bh
	Port P2 direction	P2DIR	02Ah
	Port P2 output	P2OUT	029h
	Port P2 input	P2IN	028h
Port P1	Port P1 selection	P1SEL	026h
	Port P1 interrupt enable	P1IE	025h
	Port P1 interrupt edge select	P1IES	024h
	Port P1 interrupt flag	P1IFG	023h
	Port P1 direction	P1DIR	022h
	Port P1 output	P1OUT	021h
	Port P1 input	P1IN	020h
Special Function	SFR interrupt flag 2	IFG2	003h
	SFR interrupt flag 1	IFG1	002h
	SFR interrupt enable 2	IE2	001h
	SFR interrupt enable 1	IE1	000h

absolute maximum ratings†

Voltage applied at V_{CC} to V_{SS} ..... −0.3 V to 4.1 V
Voltage applied to any pin (see Note) ..... −0.3 V to V_{CC}+0.3 V
Diode current at any device terminal ..... ±2 mA
Storage temperature, T_{stg} (unprogrammed device) ..... −55°C to 150°C
Storage temperature, T_{stg} (programmed device) ..... −40°C to 85°C

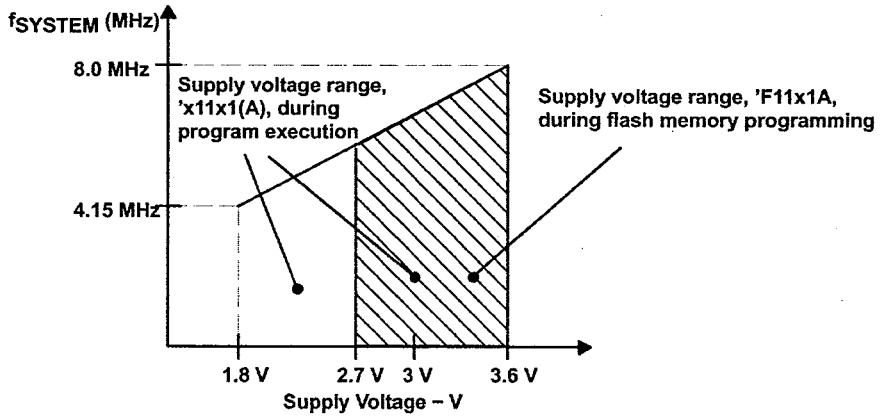
† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE: All voltages referenced to V_{SS}. The JTAG fuse-blow voltage, V_{FB}, is allowed to exceed the absolute maximum rating. The voltage is applied to the TEST pin when blowing the JTAG fuse.

recommended operating conditions

			MIN	NOM	MAX	UNITS
Supply voltage during program execution, V_{CC} (see Note 1)		MSP430C11x1	1.8		3.6	V
		MSP430F11x1A	1.8		3.6	
Supply voltage during program/erase flash memory, V_{CC}		MSP430F11x1A	2.7		3.6	V
Supply voltage, V_{SS}				0		V
Operating free-air temperature range, T_A		MSP430x11x1(A)	−40		85	°C
LFXT1 crystal frequency, $f_{(LFXT1)}$ (see Note 1 & 2)	LF mode selected, XTS=0	Watch crystal		32768		Hz
	XT1 mode selected, XTS=1	Ceramic resonator	450		8000	kHz
		Crystal	1000		8000	
Processor frequency $f_{(system)}$ (MCLK signal)		V_{CC} = 1.8 V, MSP430x11x1(A)	dc		4.15	MHz
		V_{CC} = 3.6 V, MSP430x11x1(A)	dc		8	

NOTES: 1. In LF mode, the LFXT1 oscillator requires a watch crystal. A 5.1MΩ resistor from XOUT to V_{SS} is recommended when V_{CC} < 2.5 V. In XT1 mode, the LFXT1 and XT2 oscillators accept a ceramic resonator or crystal up to 4.15MHz at $V_{CC} \geq 2.2$ V. In XT1 mode, the LFXT1 and XT2 oscillators accept a ceramic resonator or crystal up to 8MHz at $V_{CC} \geq 2.8$ V.
2. In LF mode, the LFXT1 oscillator requires a watch crystal. In XT1 mode, LFXT1 accepts a ceramic resonator or a crystal.



NOTE: Minimum processor frequency is defined by system clock. Flash program or erase operations require a minimum V_{CC} of 2.7 V.

Figure 1. Frequency vs Supply Voltage, MSP430x11x1(A)

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

supply current (into V_{CC}) excluding external current

PARAMETER			TEST CONDITIONS			MIN	TYP	MAX	UNIT
$I_{(AM)}$	Active mode	C11x1	T_A = −40°C to + 85°C, $f_{(MCLK)} = f_{(SMCLK)}$ = 1 MHz, $f_{(ACLK)}$ = 32,768 Hz		V_{CC} = 2.2 V		160	200	µA
					V_{CC} = 3 V		240	300	
			T_A = −40°C to + 85°C, $f_{(MCLK)} = f_{(SMCLK)} = f_{(ACLK)}$ = 4096 Hz		V_{CC} = 2.2 V		1.3	2	
					V_{CC} = 3 V		2.5	3.2	
		F11x1A	T_A = −40°C to + 85°C, $f_{MCLK} = f_{(SMCLK)}$ = 1 MHz, $f_{(ACLK)}$ = 32,768 Hz, Program executes in flash		V_{CC} = 2.2 V		200	250	
					V_{CC} = 3 V		300	350	
			T_A = −40°C to + 85°C, Program executes in flash $f_{(MCLK)} = f_{(SMCLK)} = f_{(ACLK)}$ = 4096 Hz		V_{CC} = 2.2 V		3	5	
					V_{CC} = 3 V		11	18	
$I_{(CPUOff)}$	Low-power mode, (LPM0)	C11x1	T_A = −40°C to + 85°C, $f_{(MCLK)}$ = 0, $f_{(SMCLK)}$ = 1 MHz, $f_{(ACLK)}$ = 32,768 Hz		V_{CC} = 2.2 V		30	40	µA
					V_{CC} = 3 V		51	60	
		F11x1A	T_A = −40°C to + 85°C, $f_{(MCLK)}$ = 0, $f_{(SMCLK)}$ = 1 MHz, $f_{(ACLK)}$ = 32,768 Hz		V_{CC} = 2.2 V		32	45	
					V_{CC} = 3 V		55	70	
$I_{(LPM2)}$	Low-power mode, (LPM2)		T_A = −40°C to + 85°C, $f_{(MCLK)} = f_{(SMCLK)}$ = 0 MHz, $f_{(ACLK)}$ = 32,768 Hz, SCG0 = 0		V_{CC} = 2.2 V		11	14	µA
					V_{CC} = 3 V		17	22	
$I_{(LPM3)}$	Low-power mode, (LPM3)	C11x1	T_A = −40°C to + 85°C, $f_{(MCLK)} = f_{(SMCLK)}$ = 0 MHz, $f_{(ACLK)}$ = 32,768 Hz, SCG0 = 1		V_{CC} = 2.2 V		1.2	1.7	µA
					V_{CC} = 3 V		2	2.7	
		F11x1A	T_A = −40°C	$f_{(MCLK)}$ = 0 MHz, $f_{(SMCLK)}$ = 0 MHz, $f_{(ACLK)}$ = 32,768 Hz, SCG0 = 1	V_{CC} = 2.2 V,		0.8	1.2	
			T_A = 25°C				0.7	1	
			T_A = 85°C				1.6	2.3	
			T_A = −40°C		V_{CC} = 3 V		1.8	2.2	
			T_A = 25°C				1.6	1.9	
			T_A = 85°C				2.3	3.4	
$I_{(LPM4)}$	Low-power mode, (LPM4)	C11x1	T_A = −40°C	$f_{(MCLK)}$ = 0 MHz, $f_{(SMCLK)}$ = 0 MHz, $f_{(ACLK)}$ = 0 Hz, SCG0 = 1	V_{CC} = 2.2 V/3 V		0.1	0.5	µA
			T_A = 25°C				0.1	0.5	
			T_A = 85°C				0.4	0.8	
		F11x1A	T_A = −40°C		V_{CC} = 2.2 V/3 V		0.1	0.5	
			T_A = 25°C				0.1	0.5	
			T_A = 85°C				0.8	1.9	

NOTE: All inputs are tied to 0 V or V_{CC}. Outputs do not source or sink any current.

current consumption of active mode versus system frequency, C version, F version

$$I_{AM} = I_{AM[1\ MHz]} \times f_{system}\ [MHz]$$

current consumption of active mode versus supply voltage, C version

$$I_{AM} = I_{AM[3\ V]} + 105\ \mu A/V \times (V_{CC} - 3\ V)$$

current consumption of active mode versus supply voltage, F version

$$I_{AM} = I_{AM[3\ V]} + 120\ \mu A/V \times (V_{CC} - 3\ V)$$

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (continued)

Schmitt-trigger inputs – Ports P1 and P2; (P1.0 to P1.7, P2.0 to P2.5)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
V_{IT+}	Positive-going input threshold voltage	V_{CC} = 2.2 V	1.1		1.5	V
		V_{CC} = 3 V	1.5		1.9	
V_{IT-}	Negative-going input threshold voltage	V_{CC} = 2.2 V	0.4		0.9	V
		V_{CC} = 3 V	0.9		1.3	
V_{hys}	Input voltage hysteresis ($V_{IT+} - V_{IT-}$)	V_{CC} = 2.2 V	0.3		1.1	V
		V_{CC} = 3 V	0.5		1	

standard inputs – $\overline{\text{RST}}$/NMI, JTAG: TCK, TMS, TDI/TCLK

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
V_{IL}	Low-level input voltage	V_{CC} = 2.2 V / 3 V	V_{SS}		V_{SS}+0.6	V
V_{IH}	High-level input voltage		0.8×V_{CC}		V_{CC}	V

inputs Px.x, TAx

PARAMETER		TEST CONDITIONS	V_{CC}	MIN	TYP	MAX	UNIT
$t_{(int)}$	External interrupt timing	Port P1, P2: P1.x to P2.x, External trigger signal for the interrupt flag, (see Note 1)	2.2 V/3 V	1.5			cycle
			2.2 V	62			ns
			3 V	50			
$t_{(cap)}$	Timer_A, capture timing	TA0, TA1, TA2	2.2 V	62			ns
			3 V	50			
$f_{(TAext)}$	Timer_A clock frequency externally applied to pin	TACLK, INCLK $t_{(H)} = t_{(L)}$	2.2 V			8	MHz
			3 V			10	
$f_{(TAint)}$	Timer_A clock frequency	SMCLK or ACLK signal selected	2.2 V			8	MHz
			3 V			10	

NOTES: 1. The external signal sets the interrupt flag every time the minimum $t_{(int)}$ cycle and time parameters are met. It may be set even with trigger signals shorter than $t_{(int)}$. Both the cycle and timing specifications must be met to ensure the flag is set. $t_{(int)}$ is measured in MCLK cycles.

leakage current

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
$I_{lkg(Px.x)}$	High-impedance leakage current	Port P1: P1.x, 0 ≤ x ≤ 7 (see Notes 1, 2)	V_{CC} = 2.2 V/3 V,			±50	nA
		Port P2: P2.x, 0 ≤ x ≤ 5 (see Notes 1, 2)	V_{CC} = 2.2 V/3 V,			±50	

NOTES: 1. The leakage current is measured with V_{SS} or V_{CC} applied to the corresponding pin(s), unless otherwise noted.
2. The leakage of the digital port pins is measured individually. The port pin must be selected for input and there must be no optional pullup or pulldown resistor.

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (continued)

outputs – Ports P1 and P2; (P1.0 to P1.7, P2.0 to P2.5)

PARAMETER		TEST CONDITIONS			MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage Port 1 and Port 2 (C11x1) Port 1 (F11x1A)	$I_{(OHmax)}$ = −1.5 mA	V_{CC} = 2.2 V	See Note 1	V_{CC}−0.25		V_{CC}	V
		$I_{(OHmax)}$ = −6 mA		See Note 2	V_{CC}−0.6		V_{CC}	
		$I_{(OHmax)}$ = −1.5 mA	V_{CC} = 3 V	See Note 1	V_{CC}−0.25		V_{CC}	
		$I_{(OHmax)}$ = −6 mA		See Note 2	V_{CC}−0.6		V_{CC}	
V_{OH}	High-level output voltage Port 2 (F11x1A)	$I_{(OHmax)}$ = −1 mA	V_{CC} = 2.2 V	See Note 3	V_{CC}−0.25		V_{CC}	V
		$I_{(OHmax)}$ = −3.4 mA		See Note 3	V_{CC}−0.6		V_{CC}	
		$I_{(OHmax)}$ = −1 mA	V_{CC} = 3 V	See Note 3	V_{CC}−0.25		V_{CC}	
		$I_{(OHmax)}$ = −3.4 mA		See Note 3	V_{CC}−0.6		V_{CC}	
V_{OL}	Low-level output voltage Port 1 and Port 2 (C11x1, F11x1A)	$I_{(OLmax)}$ = 1.5 mA	V_{CC} = 2.2 V	See Note 1	V_{SS}		V_{SS}+0.25	V
		$I_{(OLmax)}$ = 6 mA		See Note 2	V_{SS}		V_{SS}+0.6	
		$I_{(OLmax)}$ = 1.5 mA	V_{CC} = 3 V	See Note 1	V_{SS}		V_{SS}+0.25	
		$I_{(OLmax)}$ = 6 mA		See Note 2	V_{SS}		V_{SS}+0.6	

NOTES: 1. The maximum total current, I_{OHmax} and I_{OLmax}, for all outputs combined, should not exceed ±12 mA to hold the maximum voltage drop specified.
2. The maximum total current, I_{OHmax} and I_{OLmax}, for all outputs combined, should not exceed ±48 mA to hold the maximum voltage drop specified.
3. One output loaded at a time.

output frequency

PARAMETER		TEST CONDITIONS		VCC	MIN	TYP	MAX	UNIT
f_{P20}	Output frequency	P2.0/ACLK, C_L = 20 pF		2.2 V/3 V			f_{System}	MHz
f_{TAx}		TA0, TA1, TA2, C_L = 20 pF Internal clock source, SMCLK signal applied (see Note 1)		2.2 V/3 V	dc		f_{System}	
t_{Xdc}	Duty cycle of O/P frequency	P1.4/SMCLK, C_L = 20 pF	$f_{SMCLK} = f_{LFXT1} = f_{XT1}$	2.2 V/3 V	40%		60%	
			$f_{SMCLK} = f_{LFXT1} = f_{LF}$		35%		65%	
			$f_{SMCLK} = f_{LFXT1/n}$		50%− 15 ns	50%	50%+ 15 ns	
			$f_{SMCLK} = f_{DCOCLK}$	2.2 V/3 V	50%− 15 ns	50%	50%+ 15 ns	
		P2.0/ACLK, C_L = 20 pF	$f_{P20} = f_{LFXT1} = f_{XT1}$	2.2 V/3 V	40%		60%	
			$f_{P20} = f_{LFXT1} = f_{LF}$		30%		70%	
			$f_{P20} = f_{LFXT1/n}$			50%		
t_{TAdc}		TA0, TA1, TA2, C_L = 20 pF, duty cycle = 50%		2.2 V/3 V		0	±50	ns

NOTE 1: The limits of the system clock MCLK has to be met. MCLK and SMCLK can have different frequencies.

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (continued)

outputs – Ports P1 and P2 (continued)

TYPICAL LOW-LEVEL OUTPUT CURRENT
vs
LOW-LEVEL OUTPUT VOLTAGE

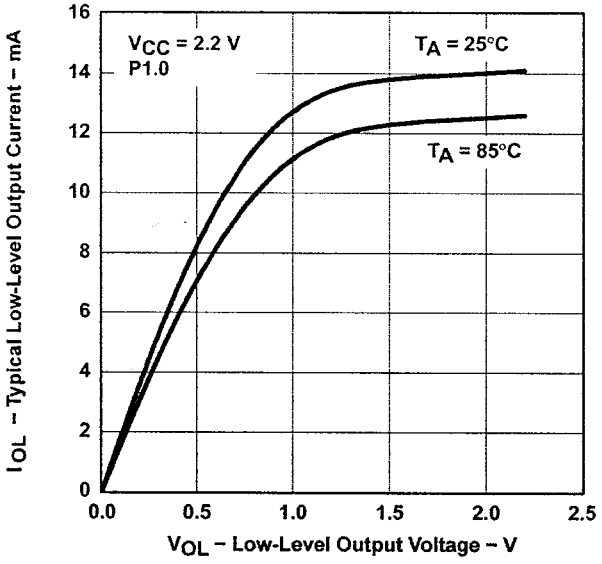


Figure 2

TYPICAL LOW-LEVEL OUTPUT CURRENT
vs
LOW-LEVEL OUTPUT VOLTAGE

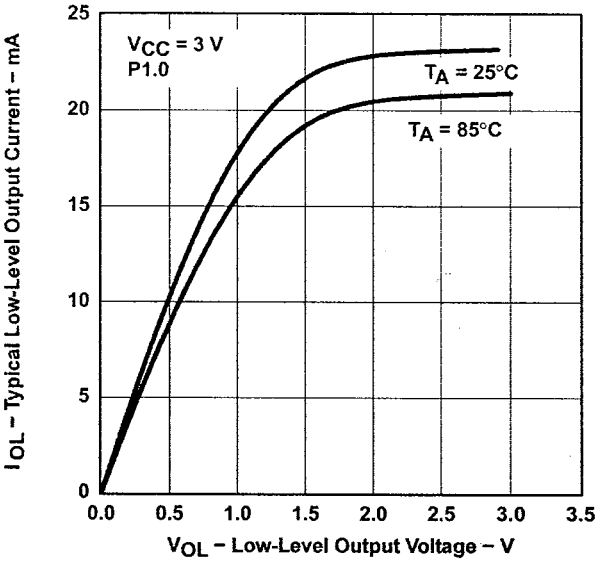


Figure 3

TYPICAL HIGH-LEVEL OUTPUT CURRENT
vs
HIGH-LEVEL OUTPUT VOLTAGE

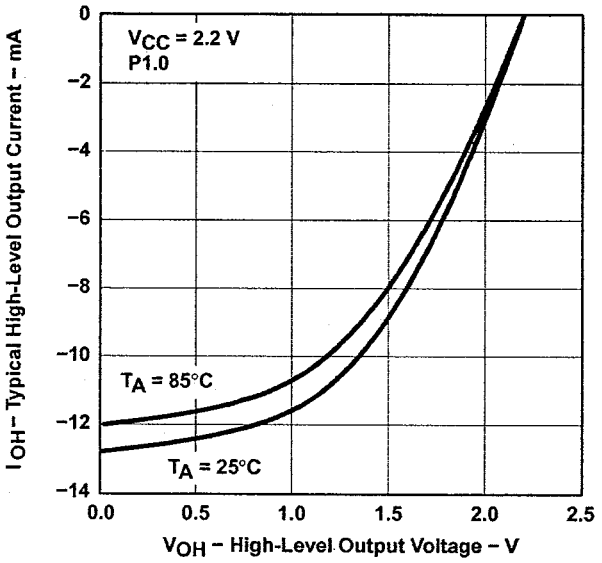


Figure 4

TYPICAL HIGH-LEVEL OUTPUT CURRENT
vs
HIGH-LEVEL OUTPUT VOLTAGE

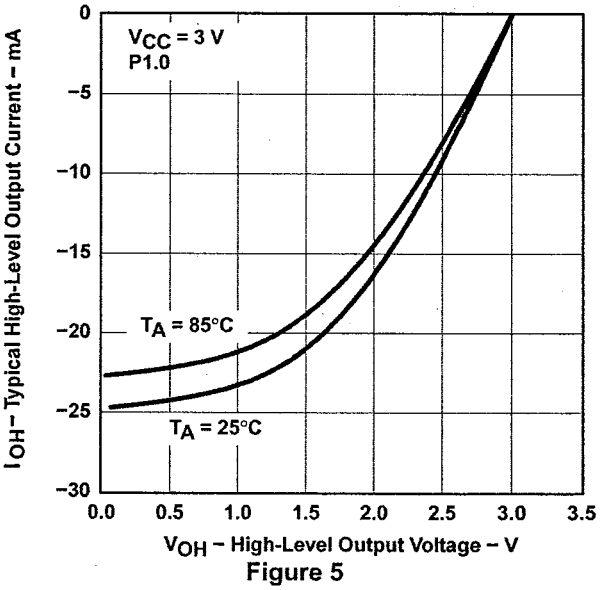


Figure 5

NOTE: One output loaded at a time.

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (continued)

optional resistors, individually programmable with ROM code (see Note 1)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
$R_{(opt1)}$	Resistors, individually programmable with ROM code, all port pins, values applicable for pulldown and pullup	V_{CC} = 2.2 V/3 V	2.5	5	10	kΩ
$R_{(opt2)}$			3.8	7.7	15	kΩ
$R_{(opt3)}$			7.6	15	31	kΩ
$R_{(opt4)}$			11.5	23	46	kΩ
$R_{(opt5)}$			23	45	90	kΩ
$R_{(opt6)}$			46	90	180	kΩ
$R_{(opt7)}$			70	140	280	kΩ
$R_{(opt8)}$			115	230	460	kΩ
$R_{(opt9)}$			160	320	640	kΩ
$R_{(opt10)}$			205	420	830	kΩ

NOTE 1: Optional resistors R_{Optx} for pulldown or pullup are not available in standard flash memory device MSP430F11x1A.

wake-up from lower power modes (LPMx)

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
$t_{(LPM0)}$	Delay time (see Note 1)	V_{CC} = 2.2 V/3 V			100		ns
$t_{(LPM2)}$		V_{CC} = 2.2 V/3 V			100		ns
$t_{(LPM3)}$		$f_{(MCLK)}$ = 1 MHz,	V_{CC} = 2.2 V/3 V			6	µs
		$f_{(MCLK)}$ = 2 MHz,	V_{CC} = 2.2 V/3 V			6	µs
		$f_{(MCLK)}$ = 3 MHz,	V_{CC} = 2.2 V/3 V			6	µs
$t_{(LPM4)}$		$f_{(MCLK)}$ = 1 MHz,	V_{CC} = 2.2 V/3 V			6	µs
		$f_{(MCLK)}$ = 2 MHz,	V_{CC} = 2.2 V/3 V			6	µs
		$f_{(MCLK)}$ = 3 MHz,	V_{CC} = 2.2 V/3 V			6	µs

NOTE 1: Parameter applicable only if DCOCLK is used for MCLK.

RAM

PARAMETER		MIN	NOM	MAX	UNIT
$V_{(RAMh)}$	CPU halted (see Note 1)	1.6			V

NOTE 1: This parameter defines the minimum supply voltage V_{CC} when the data in the program memory RAM remains unchanged. No program execution should happen during this supply voltage condition.

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (continued)

Comparator_A (see Note 1)

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
$I_{(DD)}$		CAON=1, CARSEL=0, CAREF=0	V_{CC} = 2.2 V		25	40	µA
			V_{CC} = 3 V		45	60	
$I_{(Refladder/RefDiode)}$		CAON=1, CARSEL=0, CAREF=1/2/3, no load at P2.3/CA0/TA1 and P2.4/CA1/TA2	V_{CC} = 2.2 V		30	50	µA
			V_{CC} = 3 V		45	71	
$V_{(IC)}$	Common-mode input voltage	CAON =1	V_{CC} = 2.2 V/3 V	0		V_{CC}–1	V
$V_{(Ref025)}$	Voltage @ 0.25 V_{CC} node / V_{CC}	PCA0=1, CARSEL=1, CAREF=1, No load at P2.3/CA0/TA1 and P2.4/CA1/TA2	V_{CC} = 2.2 V/3 V	0.23	0.24	0.25	
$V_{(Ref050)}$	Voltage @ 0.5V_{CC} node / V_{CC}	PCA0=1, CARSEL=1, CAREF=2, No load at P2.3/CA0/TA1 and P2.4/CA1/TA2	V_{CC} = 2.2 V/3 V	0.47	0.48	0.5	
$V_{(RefVT)}$	(see Figure 6 and Figure 7)	PCA0=1, CARSEL=1, CAREF=3, No load at P2.3/CA0/TA1 and P2.4/CA1/TA2, T_A = 85°C	V_{CC} = 2.2 V	390	480	540	mV
			V_{CC} = 3 V	400	490	550	
$V_{(offset)}$	Offset voltage	See Note 2	V_{CC} = 2.2 V/3 V	–30		30	mV
V_{hys}	Input hysteresis	CAON=1	V_{CC} = 2.2 V/3 V	0	0.7	1.4	mV
$t_{(response\ LH)}$		T_A = 25°C, Overdrive 10 mV, Without filter: CAF=0	V_{CC} = 2.2 V	160	210	300	ns
			V_{CC} = 3 V	90	150	240	
		T_A = 25°C, Overdrive 10 mV, With filter: CAF=1	V_{CC} = 2.2 V	1.4	1.9	3.4	µs
			V_{CC} = 3 V	0.9	1.5	2.6	
$t_{(response\ HL)}$		T_A = 25°C, Overdrive 10 mV, Without filter: CAF=0	V_{CC} = 2.2 V	130	210	300	ns
			V_{CC} = 3 V	80	150	240	
		T_A = 25°C, Overdrive 10 mV, With filter: CAF=1	V_{CC} = 2.2 V	1.4	1.9	3.4	µs
			V_{CC} = 3 V	0.9	1.5	2.6	

NOTES: 1. The leakage current for the Comparator_A terminals is identical to $I_{lkg(Px.x)}$ specification.
2. The input offset voltage can be cancelled by using the CAEX bit to invert the Comparator_A inputs on successive measurements. The two successive measurements are then summed together.

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (continued)

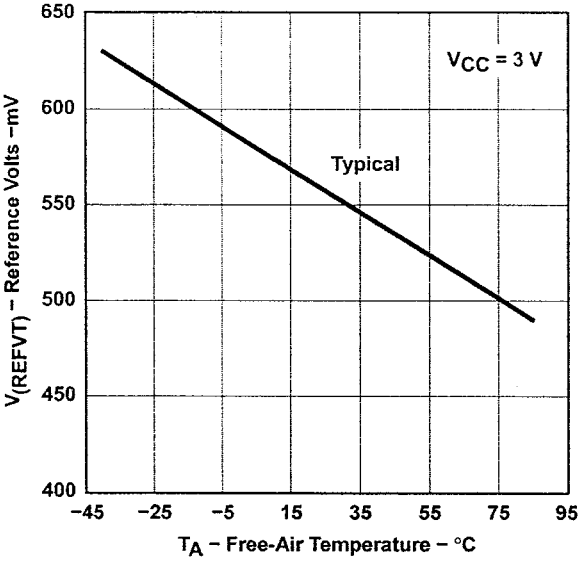


Figure 6. $V_{(RefVT)}$ vs Temperature, V_{CC} = 3 V

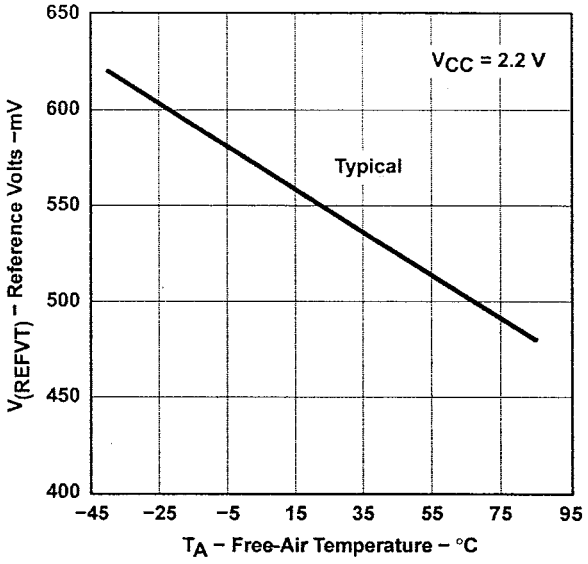


Figure 7. $V_{(RefVT)}$ vs Temperature, V_{CC} = 2.2 V

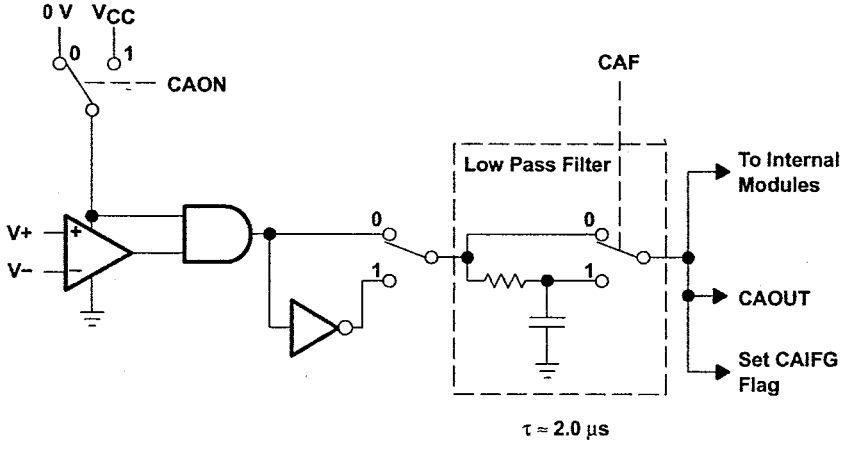


Figure 8. Block Diagram of Comparator_A Module

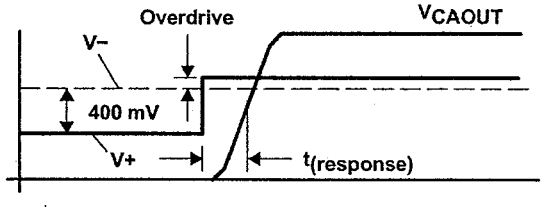


Figure 9. Overdrive Definition

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (continued)

PUC/POR

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
$t_{(POR_Delay)}$	Internal time delay to release POR		V_{CC} = 2.2 V/3 V		150	250	µs
V_{POR}	V_{CC} threshold at which POR release delay time begins (see Note 1)	T_A = −40°C		1.4		1.8	V
		T_A = 25°C		1.1		1.5	V
		T_A = 85°C		0.8		1.2	V
$V_{(min)}$	V_{CC} threshold required to generate a POR (see Note 2)	V_{CC} |dV/dt| ≥ 1V/ms		0.2			V
$t_{(reset)}$	$\overline{RST}$/NMI low time for PUC/POR	Reset is accepted internally		2			µs

NOTES: 1. V_{CC} rise time dV/dt ≥ 1V/ms.
2. When driving V_{CC} low in order to generate a POR condition, V_{CC} should be driven to 200mV or lower with a dV/dt equal to or less than −1V/ms. The corresponding rising V_{CC} must also meet the dV/dt requirement equal to or greater than +1V/ms.

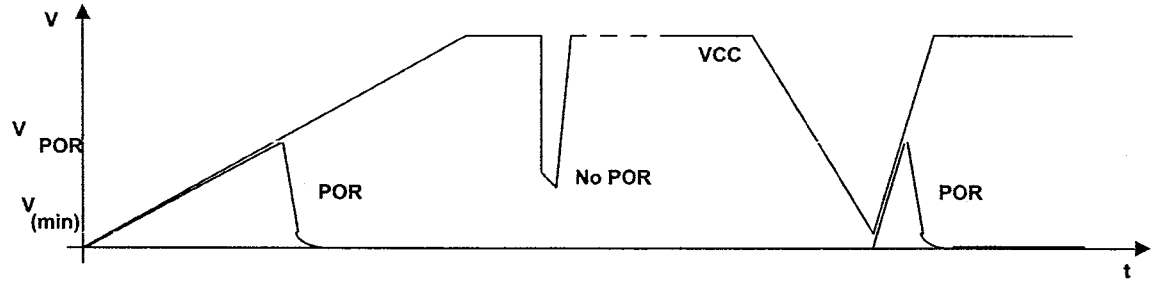


Figure 10. Power-On Reset (POR) vs Supply Voltage

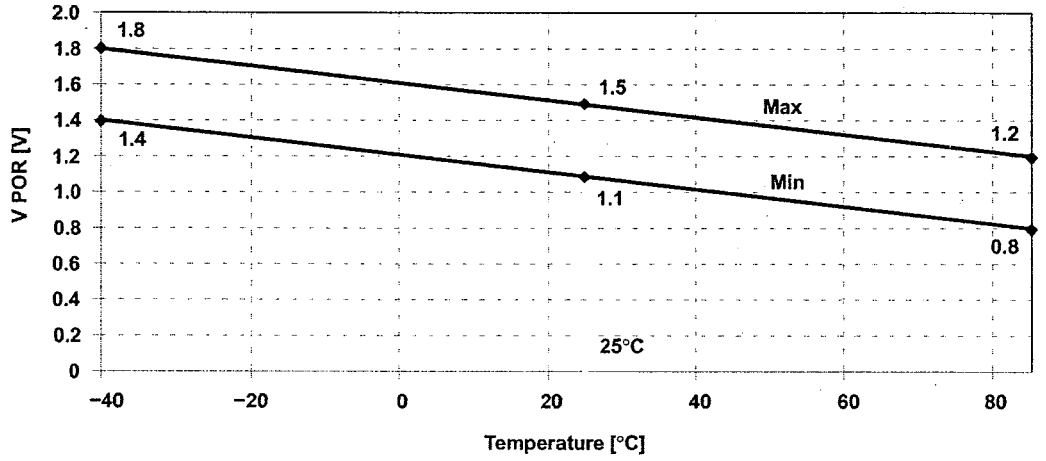


Figure 11. V_{POR} vs Temperature

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (continued)

DCO

PARAMETER	TEST CONDITIONS		MIN	TYP	MAX	UNIT
$f_{(DCO03)}$	R_{sel} = 0, DCO = 3, MOD = 0, DCOR = 0, T_A = 25°C	V_{CC} = 2.2 V	0.08	0.12	0.15	MHz
		V_{CC} = 3 V	0.08	0.13	0.16	
$f_{(DCO13)}$	R_{sel} = 1, DCO = 3, MOD = 0, DCOR = 0, T_A = 25°C	V_{CC} = 2.2 V	0.14	0.19	0.23	MHz
		V_{CC} = 3 V	0.14	0.18	0.22	
$f_{(DCO23)}$	R_{sel} = 2, DCO = 3, MOD = 0, DCOR = 0, T_A = 25°C	V_{CC} = 2.2 V	0.22	0.30	0.36	MHz
		V_{CC} = 3 V	0.22	0.28	0.34	
$f_{(DCO33)}$	R_{sel} = 3, DCO = 3, MOD = 0, DCOR = 0, T_A = 25°C	V_{CC} = 2.2 V	0.37	0.49	0.59	MHz
		V_{CC} = 3 V	0.37	0.47	0.56	
$f_{(DCO43)}$	R_{sel} = 4, DCO = 3, MOD = 0, DCOR = 0, T_A = 25°C	V_{CC} = 2.2 V	0.61	0.77	0.93	MHz
		V_{CC} = 3 V	0.61	0.75	0.9	
$f_{(DCO53)}$	R_{sel} = 5, DCO = 3, MOD = 0, DCOR = 0, T_A = 25°C	V_{CC} = 2.2 V	1	1.2	1.5	MHz
		V_{CC} = 3 V	1	1.3	1.5	
$f_{(DCO63)}$	R_{sel} = 6, DCO = 3, MOD = 0, DCOR = 0, T_A = 25°C	V_{CC} = 2.2 V	1.6	1.9	2.2	MHz
		V_{CC} = 3 V	1.69	2	2.29	
$f_{(DCO73)}$	R_{sel} = 7, DCO = 3, MOD = 0, DCOR = 0, T_A = 25°C	V_{CC} = 2.2 V	2.4	2.9	3.4	MHz
		V_{CC} = 3 V	2.7	3.2	3.65	
$f_{(DCO77)}$	R_{sel} = 7, DCO = 7, MOD = 0, DCOR = 0, T_A = 25°C	V_{CC} = 2.2 V	4	4.5	4.9	MHz
		V_{CC} = 3 V	4.4	4.9	5.4	
$f_{(DCO47)}$	R_{sel} = 4, DCO = 7, MOD = 0, DCOR = 0, T_A = 25°C	V_{CC} = 2.2 V/3 V	f_{DCO40} x1.7	f_{DCO40} x2.1	f_{DCO40} x2.5	MHz
$S_{(Rsel)}$	$S_R = f_{Rsel+1}/f_{Rsel}$	V_{CC} = 2.2 V/3 V	1.35	1.65	2	ratio
$S_{(DCO)}$	$S_{DCO} = f_{DCO+1}/f_{DCO}$	V_{CC} = 2.2 V/3 V	1.07	1.12	1.16	
D_t	Temperature drift, R_{sel} = 4, DCO = 3, MOD = 0 (see Note 1)	V_{CC} = 2.2 V	−0.31	−0.36	−0.40	%/°C
		V_{CC} = 3 V	−0.33	−0.38	−0.43	
D_V	Drift with V_{CC} variation, R_{sel} = 4, DCO = 3, MOD = 0 (see Note 1)	V_{CC} = 2.2 V/3 V	0	5	10	%/V

NOTE 1: These parameters are not production tested.

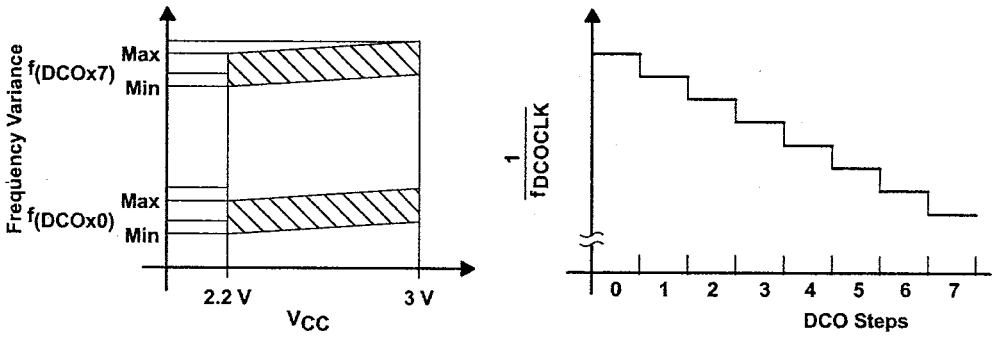


Figure 12. DCO Characteristics

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (continued)

main DCO characteristics

- Individual devices have a minimum and maximum operation frequency. The specified parameters for $f_{(DCOx0)}$ to $f_{(DCOx7)}$ are valid for all devices.
- All ranges selected by Rsel(n) overlap with Rsel(n+1): Rsel0 overlaps Rsel1, ... Rsel6 overlaps Rsel7.
- DCO control bits DCO0, DCO1, and DCO2 have a step size as defined by parameter S_{DCO}.
- Modulation control bits MOD0 to MOD4 select how often $f_{(DCO+1)}$ is used within the period of 32 DCOCLK cycles. The frequency $f_{(DCO)}$ is used for the remaining cycles. The frequency is an average equal to:

$$f_{average} = \frac{32 \times f_{(DCO)} \times f_{(DCO+1)}}{MOD \times f_{(DCO)} + (32 - MOD) \times f_{(DCO+1)}}$$

DCO when using R_{OSC} (see Note 1)

PARAMETER	TEST CONDITIONS	V_{CC}	MIN	NOM	MAX	UNIT
f_{DCO}, DCO output frequency	R_{sel} = 4, DCO = 3, MOD = 0, DCOR = 1, T_A = 25°C	2.2 V		1.8±15%		MHz
		3 V		1.95±15%		MHz
D_t, Temperature drift	R_{sel} = 4, DCO = 3, MOD = 0, DCOR = 1	2.2 V/3 V		±0.1		%/°C
D_V, Drift with V_{CC} variation	R_{sel} = 4, DCO = 3, MOD = 0, DCOR = 1	2.2 V/3 V		10		%/V

NOTES: 1. R_{OSC} = 100kΩ Metal film resistor, type 0257. 0.6 watt with 1% tolerance and T_K = ±50ppm/°C.

crystal oscillator, LFXT1

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
C_{XIN}	Input capacitance	XTS=0; LF mode selected. V_{CC} = 2.2 V / 3 V		12		pF
		XTS=1; XT1 mode selected. V_{CC} = 2.2 V / 3 V (see Note 1)		2		
C_{XOUT}	Output capacitance	XTS=0; LF mode selected. V_{CC} = 2.2 V / 3 V		12		pF
		XTS=1; XT1 mode selected. V_{CC} = 2.2 V / 3 V (see Note 1)		2		
V_{IL}	Input levels at XIN	V_{CC} = 2.2 V/3 V (see Note 2)	V_{SS}		$0.2 \times V_{CC}$	V
V_{IH}			$0.8 \times V_{CC}$		V_{CC}	

NOTES: 1. Requires external capacitors at both terminals. Values are specified by crystal manufacturers.
2. Applies only when using an external logic-level clock source. Not applicable when using a crystal or resonator.

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (continued)

Flash Memory

PARAMETER		TEST CONDITIONS	V_{CC}	MIN	NOM	MAX	UNIT
$V_{CC(PGM/ERASE)}$	Program and Erase supply voltage			2.7		3.6	V
f_{FTG}	Flash Timing Generator frequency			257		476	kHz
I_{PGM}	Supply current from V_{CC} during program		2.7 V/ 3.6 V		3	5	mA
I_{ERASE}	Supply current from V_{CC} during erase		2.7 V/ 3.6 V		3	7	mA
t_{CPT}	Cumulative program time	see Note 1	2.7 V/ 3.6 V			4	ms
$t_{CMErase}$	Cumulative mass erase time	see Note 2	2.7 V/ 3.6 V	200			ms
	Program/Erase endurance			10^4	10^5		cycles
$t_{Retention}$	Data retention duration	T_J = 25°C		100			years
t_{Word}	Word or byte program time	see Note 3			35		t_{FTG}
$t_{Block, 0}$	Block program time for 1st byte or word				30		
$t_{Block, 1-63}$	Block program time for each additional byte or word				21		
$t_{Block, End}$	Block program end-sequence wait time				6		
$t_{Mass Erase}$	Mass erase time				5297		
$t_{Seg Erase}$	Segment erase time				4819		

NOTES: 1. The cumulative program time must not be exceeded when writing to a 64-byte flash block. This parameter applies to all programming methods: individual word/byte write and block write modes.
2. The mass erase duration generated by the flash timing generator is at least 11.1ms (= 5297x1/$f_{FTG,max}$ = 5297x1/476kHz). To achieve the required cumulative mass erase time the Flash Controller's mass erase operation can be repeated until this time is met. (A worst case minimum of 19 cycles are required).
3. These values are hardwired into the Flash Controller's state machine (t_{FTG} = 1/f_{FTG}).

JTAG Interface

PARAMETER		TEST CONDITIONS	V_{CC}	MIN	NOM	MAX	UNIT
f_{TCK}	TCK input frequency	see Note 1	2.2 V	0		5	MHz
			3 V	0		10	MHz
$R_{Internal}$	Internal pull-down resistance on TEST	see Note 2	2.2 V/ 3 V	25	60	90	kΩ

NOTES: 1. f_{TCK} may be restricted to meet the timing requirements of the module selected.
2. TEST pull-down resistor implemented in all versions.

JTAG Fuse (see Note 1)

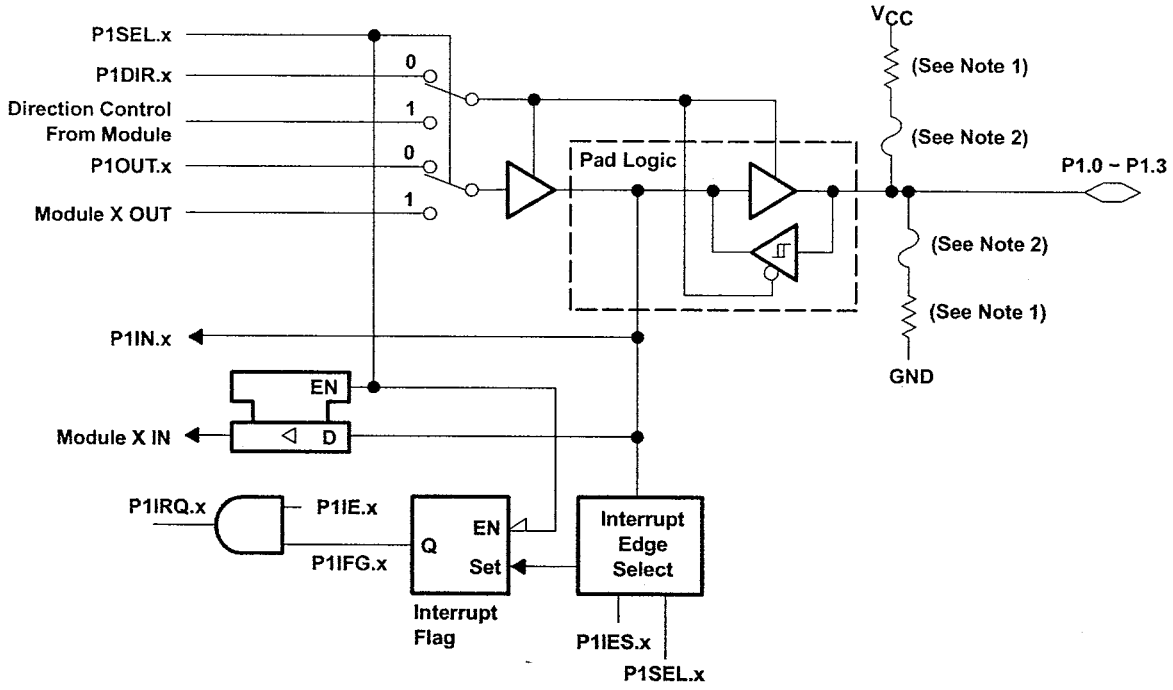
PARAMETER		TEST CONDITIONS	V_{CC}	MIN	NOM	MAX	UNIT
$V_{CC(FB)}$	Supply voltage during fuse-blow condition	T_A = 25°C		2.5			V
V_{FB}	Voltage level on TEST for fuse-blow - 'C11x1			3.5		3.9	V
	Voltage level on TEST for fuse-blow - 'F11x1A			6		7	V
I_{FB}	Supply current into TEST during fuse blow					100	mA
t_{FB}	Time to blow fuse					1	ms

NOTES: 1. Once the fuse is blown, no further access to the MSP430 JTAG/Test and emulation features is possible. The JTAG block is switched to bypass mode.

APPLICATION INFORMATION

input/output schematic

Port P1, P1.0 to P1.3, input/output with Schmitt-trigger



NOTE: x = Bit/identifier, 0 to 3 for port P1

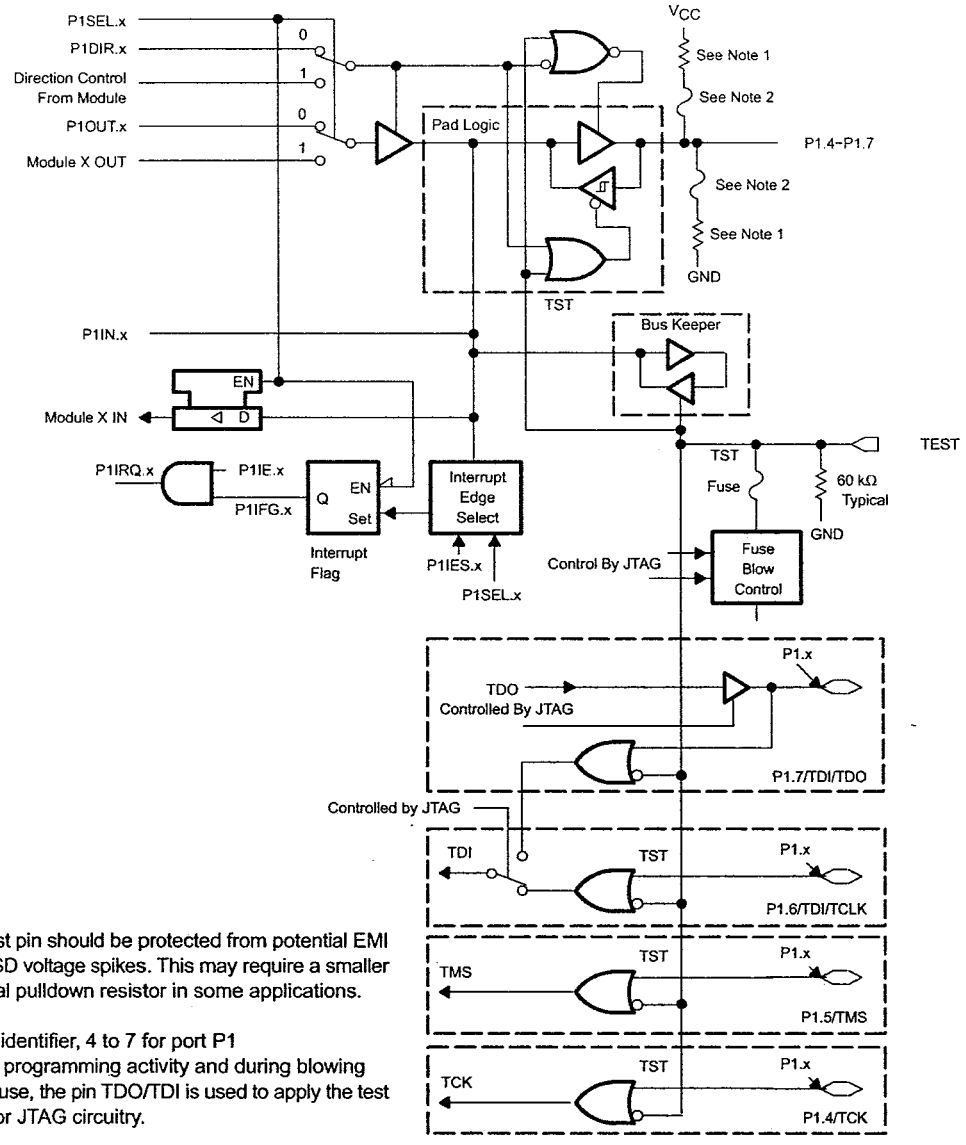
PnSel.x	PnDIR.x	Direction control from module	PnOUT.x	Module X OUT	PnIN.x	Module X IN	PnIE.x	PnIFG.x	PnIES.x
P1Sel.0	P1DIR.0	P1DIR.0	P1OUT.0	V_{SS}	P1IN.0	TACLK†	P1IE.0	P1IFG.0	P1IES.0
P1Sel.1	P1DIR.1	P1DIR.1	P1OUT.1	Out0 signal†	P1IN.1	CCI0A†	P1IE.1	P1IFG.1	P1IES.1
P1Sel.2	P1DIR.2	P1DIR.2	P1OUT.2	Out1 signal†	P1IN.2	CCI1A†	P1IE.2	P1IFG.2	P1IES.2
P1Sel.3	P1DIR.3	P1DIR.3	P1OUT.3	Out2 signal†	P1IN.3	CCI2A†	P1IE.3	P1IFG.3	P1IES.3

† Signal from or to Timer_A

NOTES: 1. Optional selection of pullup or pulldown resistors with ROM (masked) versions
2. Fuses for optional pullup and pulldown resistors can only be programmed at the factory (ROM versions only).

APPLICATION INFORMATION

Port P1, P1.4 to P1.7, input/output with Schmitt-trigger and in-system access features



NOTE: The test pin should be protected from potential EMI and ESD voltage spikes. This may require a smaller external pulldown resistor in some applications.

x = Bit identifier, 4 to 7 for port P1
During programming activity and during blowing of the fuse, the pin TDO/TDI is used to apply the test input for JTAG circuitry.

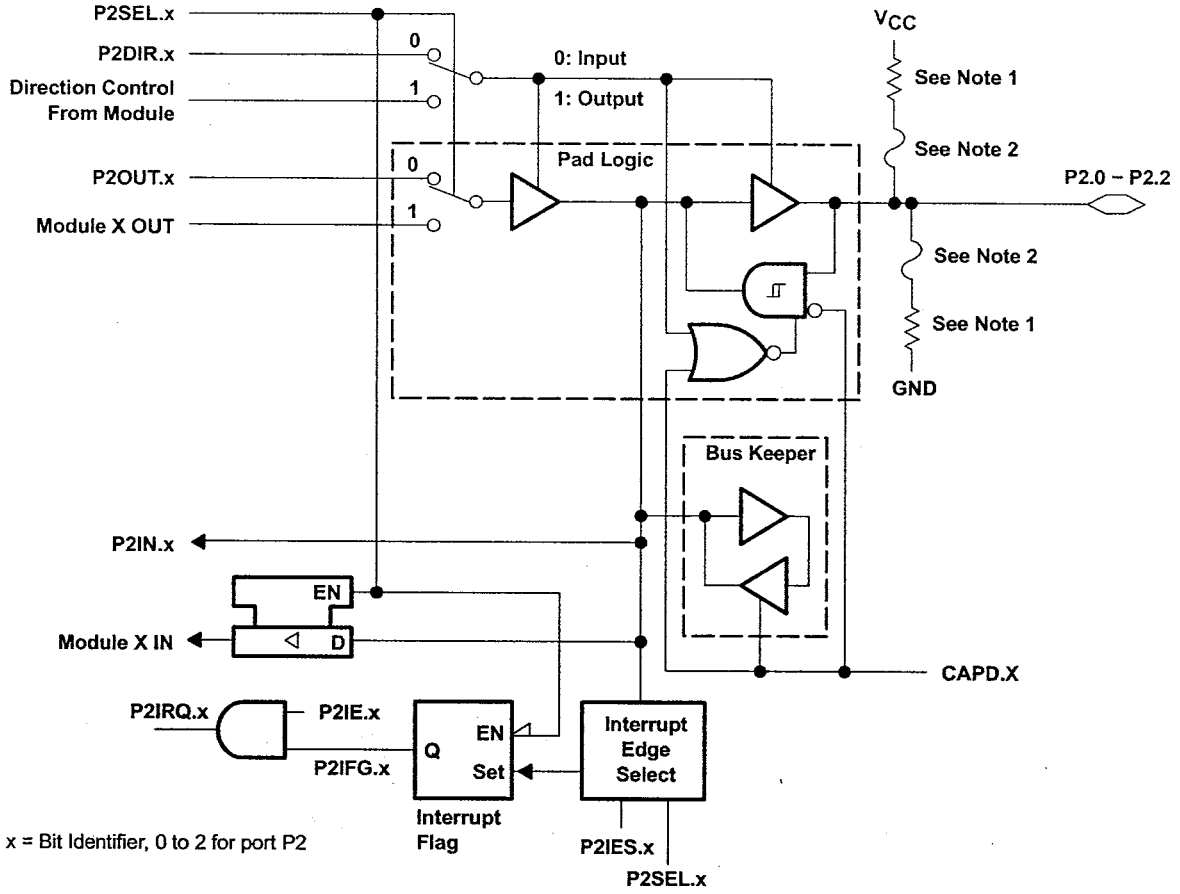
PnSel.x	PnDIR.x	Direction control from module	PnOUT.x	Module X OUT	PnIN.x	Module X IN	PnIE.x	PnIFG.x	PnIES.x
P1Sel.4	P1DIR.4	P1DIR.4	P1OUT.4	SMCLK	P1IN.4	unused	P1IE.4	P1IFG.4	P1IES.4
P1Sel.5	P1DIR.5	P1DIR.5	P1OUT.5	Out0 signal†	P1IN.5	unused	P1IE.5	P1IFG.5	P1IES.5
P1Sel.6	P1DIR.6	P1DIR.6	P1OUT.6	Out1 signal†	P1IN.6	unused	P1IE.6	P1IFG.6	P1IES.6
P1Sel.7	P1DIR.7	P1DIR.7	P1OUT.7	Out2 signal†	P1IN.7	unused	P1IE.7	P1IFG.7	P1IES.7

† Signal from or to Timer_A

NOTES: 1. Optional selection of pullup or pulldown resistors with ROM (masked) versions
2. Fuses for optional pullup and pulldown resistors can only be programmed at the factory (ROM versions only).

APPLICATION INFORMATION

Port P2, P2.0 to P2.2, input/output with Schmitt-trigger



NOTE: x = Bit Identifier, 0 to 2 for port P2

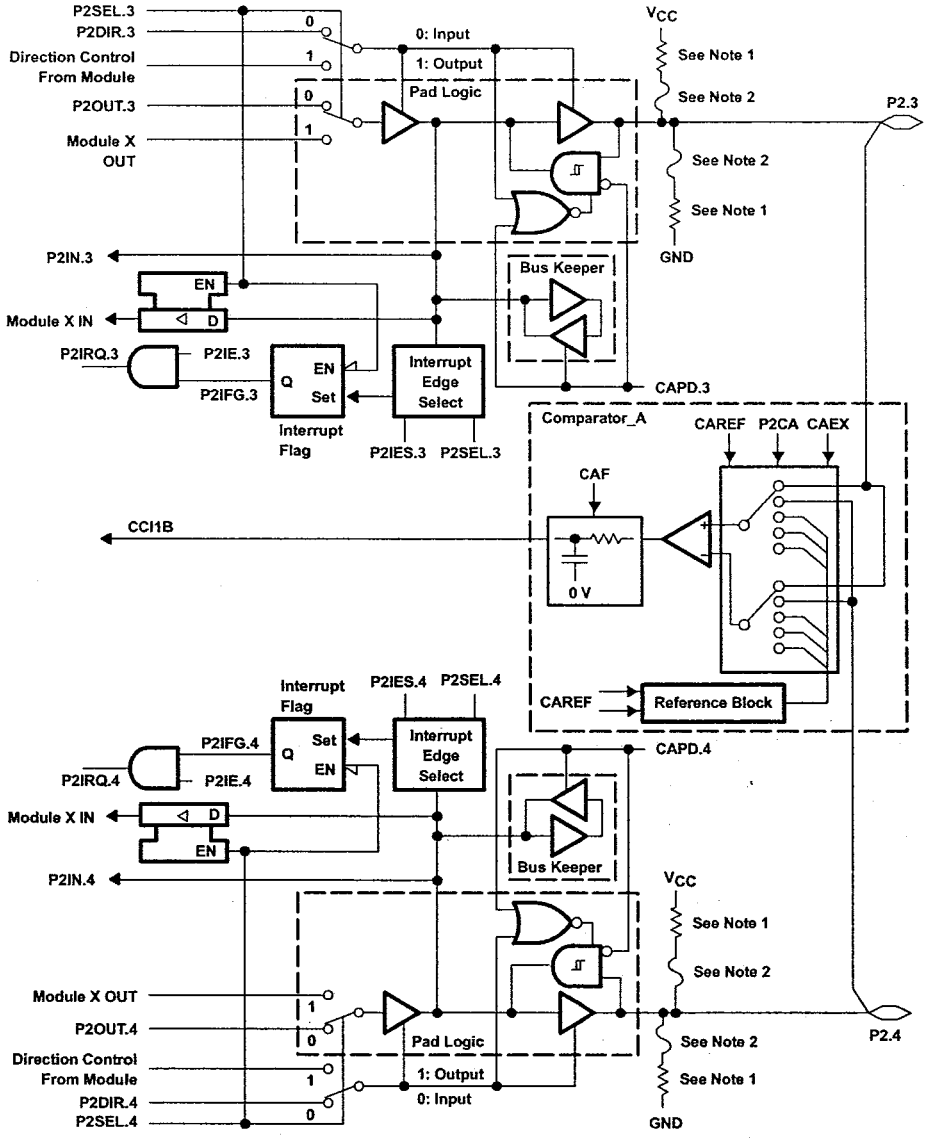
PnSel.x	PnDIR.x	Direction control from module	PnOUT.x	Module X OUT	PnIN.x	Module X IN	PnIE.x	PnIFG.x	PnIES.x
P2Sel.0	P2DIR.0	P2DIR.0	P2OUT.0	ACLK	P2IN.0	unused	P2IE.0	P2IFG.0	P1IES.0
P2Sel.1	P2DIR.1	P2DIR.1	P2OUT.1	V_{SS}	P2IN.1	INCLK†	P2IE.1	P2IFG.1	P1IES.1
P2Sel.2	P2DIR.2	P2DIR.2	P2OUT.2	CAOUT	P2IN.2	CCI0B†	P2IE.2	P2IFG.2	P1IES.2

† Signal from or to Timer_A

NOTES: 1. Optional selection of pullup or pulldown resistors with ROM (masked) versions
2. Fuses for optional pullup and pulldown resistors can only be programmed at the factory (ROM versions only).

APPLICATION INFORMATION

Port P2, P2.3 to P2.4, input/output with Schmitt-trigger



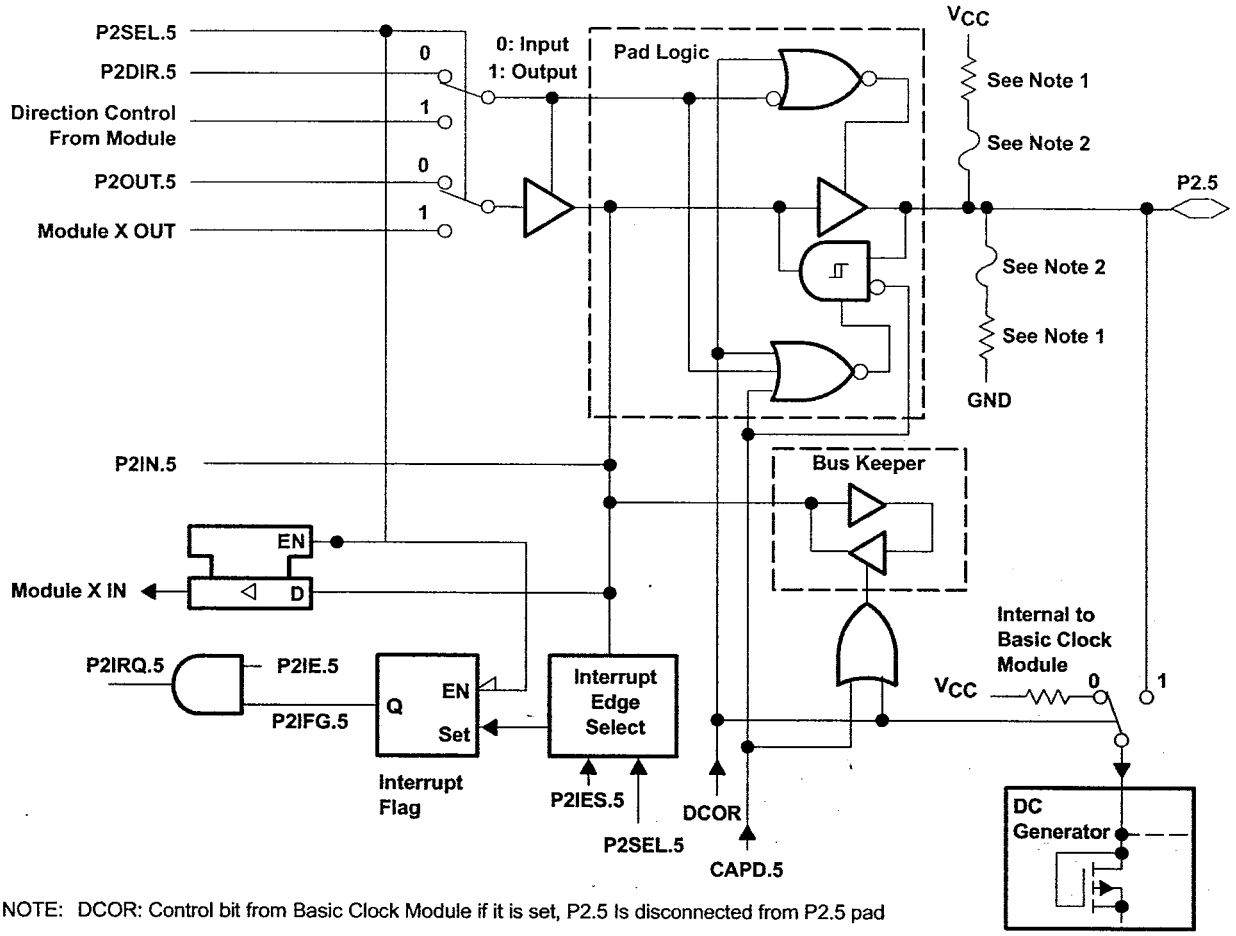
PnSel.x	PnDIR.x	Direction control from module	PnOUT.x	Module X OUT	PnIN.x	Module X IN	PnIE.x	PnIFG.x	PnIES.x
P2Sel.3	P2DIR.3	P2DIR.3	P2OUT.3	Out1 signal†	P2IN.3	unused	P2IE.3	P2IFG.3	P1IES.3
P2Sel.4	P2DIR.4	P2DIR.4	P2OUT.4	Out2 signal†	P2IN.4	unused	P2IE.4	P2IFG.4	P1IES.4

† Signal from Timer_A

NOTES: 1. Optional selection of pullup or pulldown resistors with ROM (masked) versions
2. Fuses for optional pullup and pulldown resistors can only be programmed at the factory (ROM versions only).

APPLICATION INFORMATION

Port P2, P2.5, input/output with Schmitt-trigger and R_{OSC} function for the Basic Clock module



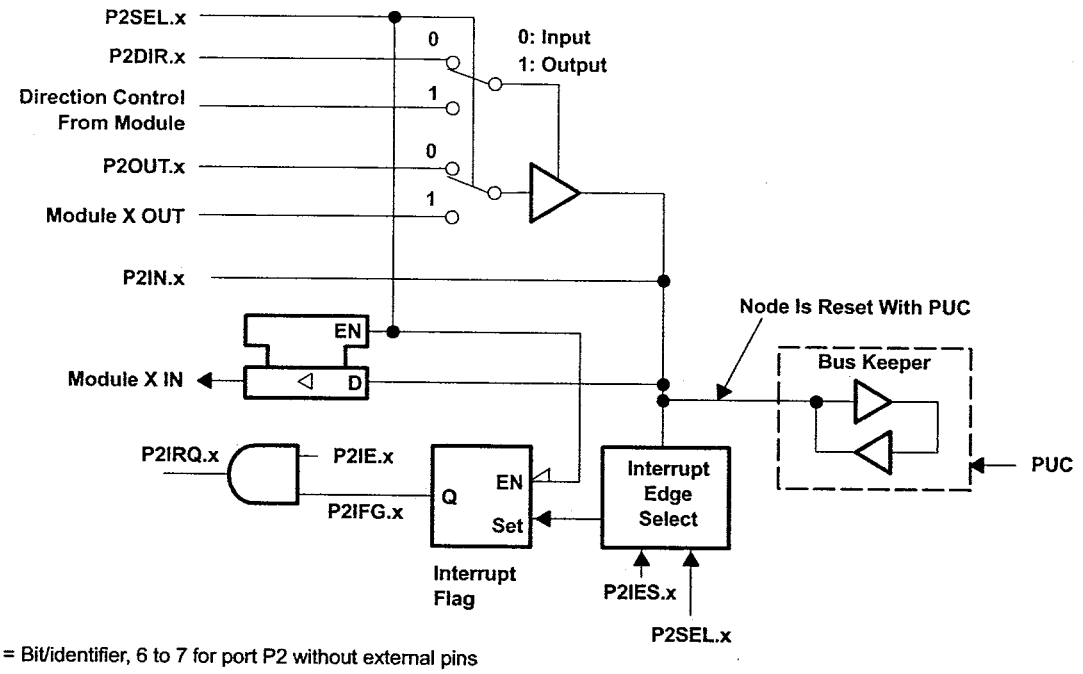
NOTE: DCOR: Control bit from Basic Clock Module if it is set, P2.5 Is disconnected from P2.5 pad

PnSel.x	PnDIR.x	Direction control from module	PnOUT.x	Module X OUT	PnIN.x	Module X IN	PnIE.x	PnIFG.x	PnIES.x
P2Sel.5	P2DIR.5	P2DIR.5	P2OUT.5	V_{SS}	P2IN.5	unused	P2IE.5	P2IFG.5	P2IES.5

NOTES: 1. Optional selection of pullup or pulldown resistors with ROM (masked) versions
2. Fuses for optional pullup and pulldown resistors can only be programmed at the factory (ROM versions only).

APPLICATION INFORMATION

Port P2, unbonded bits P2.6 and P2.7



NOTE: x = Bit/identifier, 6 to 7 for port P2 without external pins

P2Sel.x	P2DIR.x	Direction control from module	P2OUT.x	Module X OUT	P2IN.x	Module X IN	P2IE.x	P2IFG.x	P2IES.x
P2Sel.6	P2DIR.6	P2DIR.6	P2OUT.6	V_{SS}	P2IN.6	unused	P2IE.6	P2IFG.6	P2IES.6
P2Sel.7	P2DIR.7	P2DIR.7	P2OUT.7	V_{SS}	P2IN.7	unused	P2IE.7	P2IFG.7	P2IES.7

NOTE 1: Unbonded bits 6 and 7 of port P2 can be used as software interrupt flags. The interrupt flags can only be influenced by software. They work then as a software interrupt.

JTAG fuse check mode

MSP430 devices that have the fuse on the TEST terminal have a fuse check mode that tests the continuity of the fuse the first time the JTAG port is accessed after a power-on reset (POR). When activated, a fuse check current, I_{TF}, of 1 mA at 3 V, 2.5 mA at 5 V can flow from the TEST pin to ground if the fuse is not burned. Care must be taken to avoid accidentally activating the fuse check mode and increasing overall system power consumption.

When the TEST pin is taken back low after a test or programming session, the fuse check mode and sense currents are terminated.

Activation of the fuse check mode occurs with the first negative edge on the TMS pin after power up or if TMS is being held low during power up. The second positive edge on the TMS pin deactivates the fuse check mode. After deactivation, the fuse check mode remains inactive until another POR occurs. After each POR the fuse check mode has the potential to be activated.

The fuse check current will only flow when the fuse check mode is active and the TMS pin is in a low state (see Figure 13). Therefore, the additional current flow can be prevented by holding the TMS pin high (default condition).

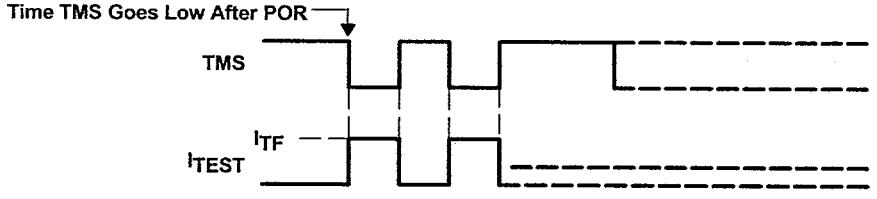


Figure 13. Fuse Check Mode Current, MSP430F11x1A and MSP430C11x1

NOTE:

The CODE and RAM data protection is ensured if the JTAG fuse is blown and the 256-bit bootloader access key is used. Also, see the *bootstrap loader* section for more information.

THERMAL PAD MECHANICAL DATA

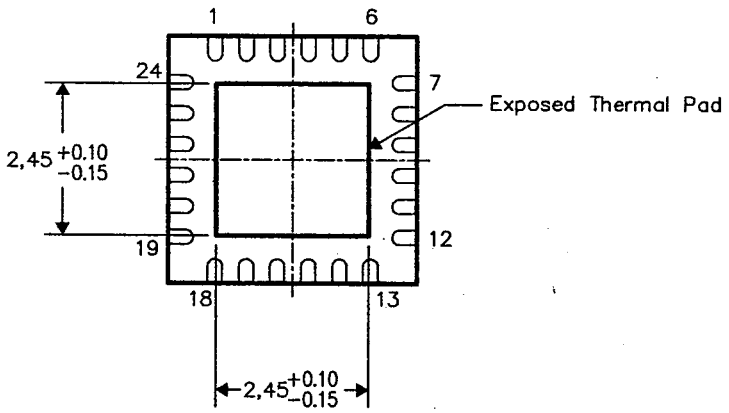
RGE (S–PQFP–N24)

THERMAL INFORMATION

This package incorporates an exposed thermal pad that is designed to be attached directly to an external heatsink. The thermal pad must be soldered directly to the printed circuit board (PCB), the PCB can be used as a heatsink. In addition, through the use of thermal vias, the thermal pad can be attached directly to a ground plane or special heatsink structure designed into the PCB. This design optimizes the heat transfer from the integrated circuit (IC).

For information on the Quad Flatpack No–Lead (QFN) package and its advantages, refer to Application Report, Quad Flatpack No–Lead Logic Packages, Texas Instruments Literature No. SCBA017. This document is available at www.ti.com.

The exposed thermal pad dimensions for this package are shown in the following illustration.



Bottom View

NOTE: All linear dimensions are in millimeters

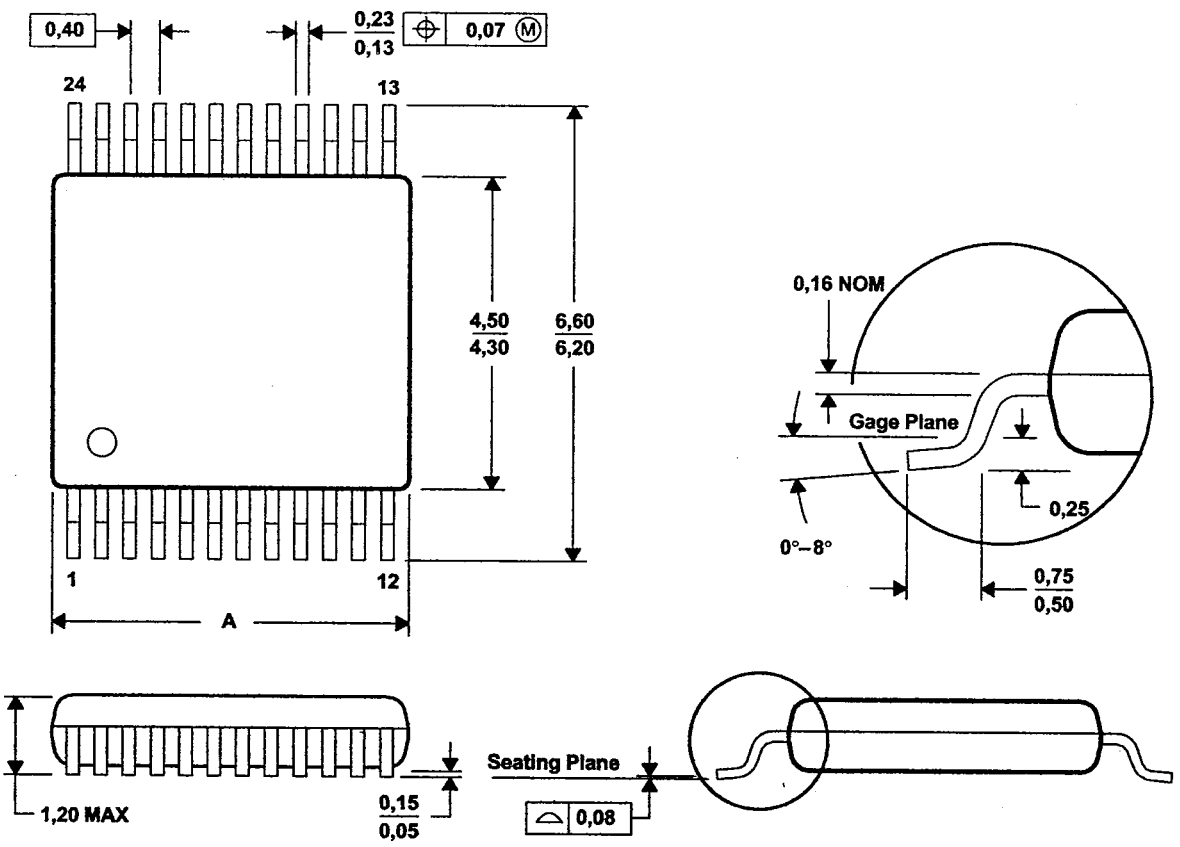
Exposed Thermal Pad Dimensions

MPDS006C – FEBRUARY 1996 – REVISED AUGUST 2000

DGV (R-PDSO-G)**

PLASTIC SMALL-OUTLINE

24 PINS SHOWN

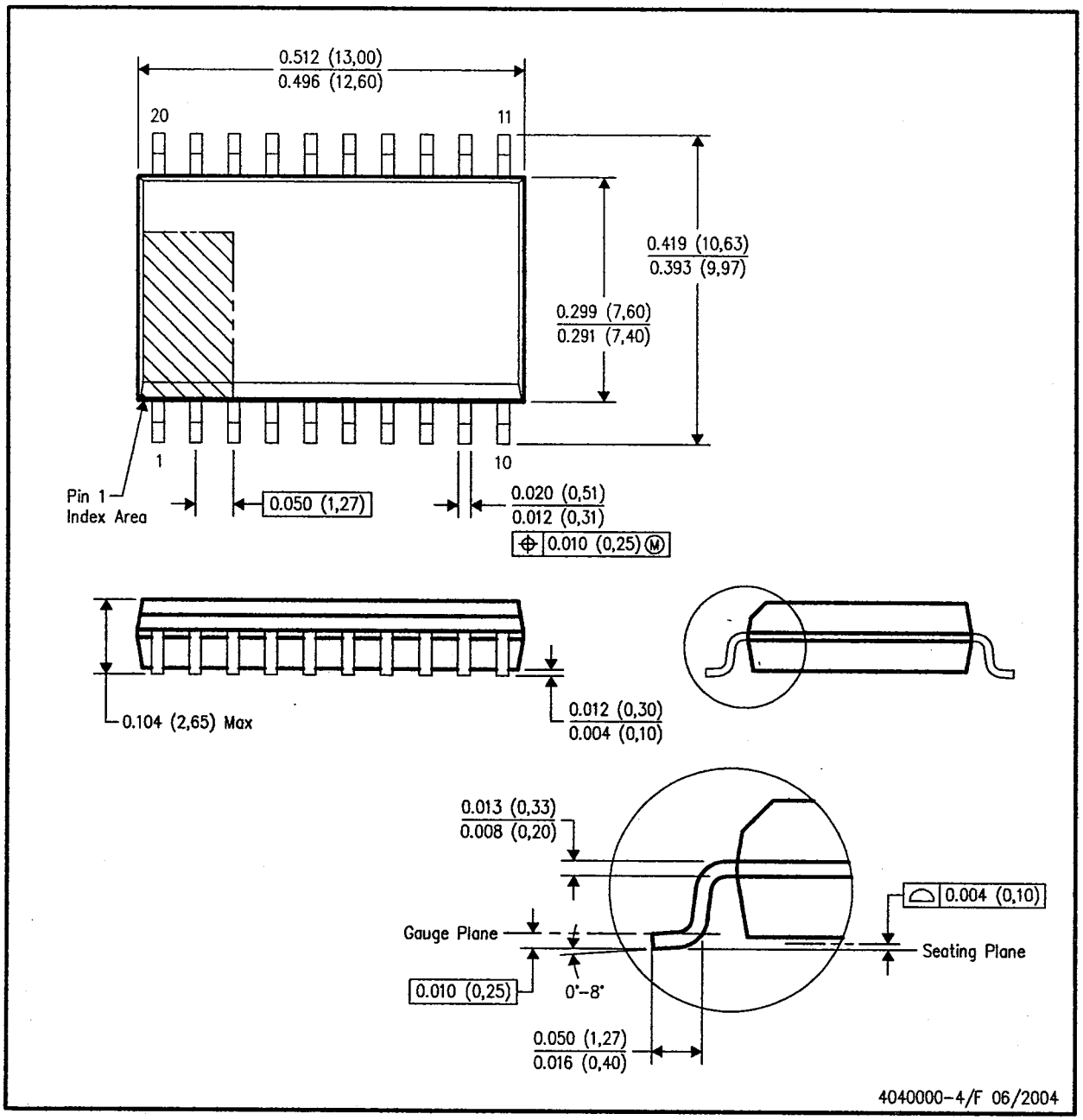


DIM \ PINS **	14	16	20	24	38	48	56
A MAX	3,70	3,70	5,10	5,10	7,90	9,80	11,40
A MIN	3,50	3,50	4,90	4,90	7,70	9,60	11,20

4073251/E 08/00

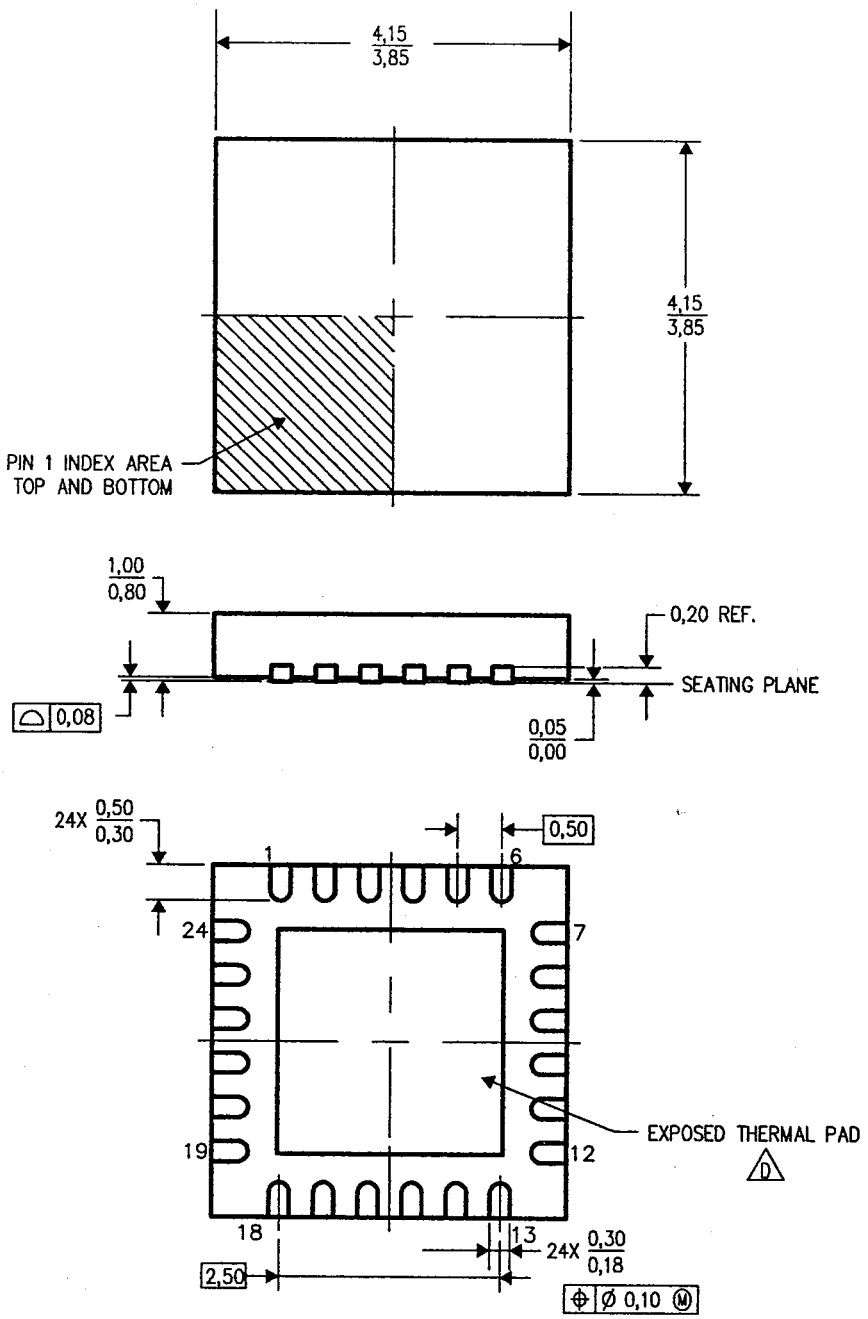
NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion, not to exceed 0,15 per side.
D. Falls within JEDEC: 24/48 Pins – MO-153
14/16/20/56 Pins – MO-194

DW (R–PDSO–G20) PLASTIC SMALL–OUTLINE PACKAGE



NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion not to exceed 0.006 (0,15).
D. Falls within JEDEC MS–013 variation AC.

RGE (S–PQFP–N24) PLASTIC QUAD FLATPACK

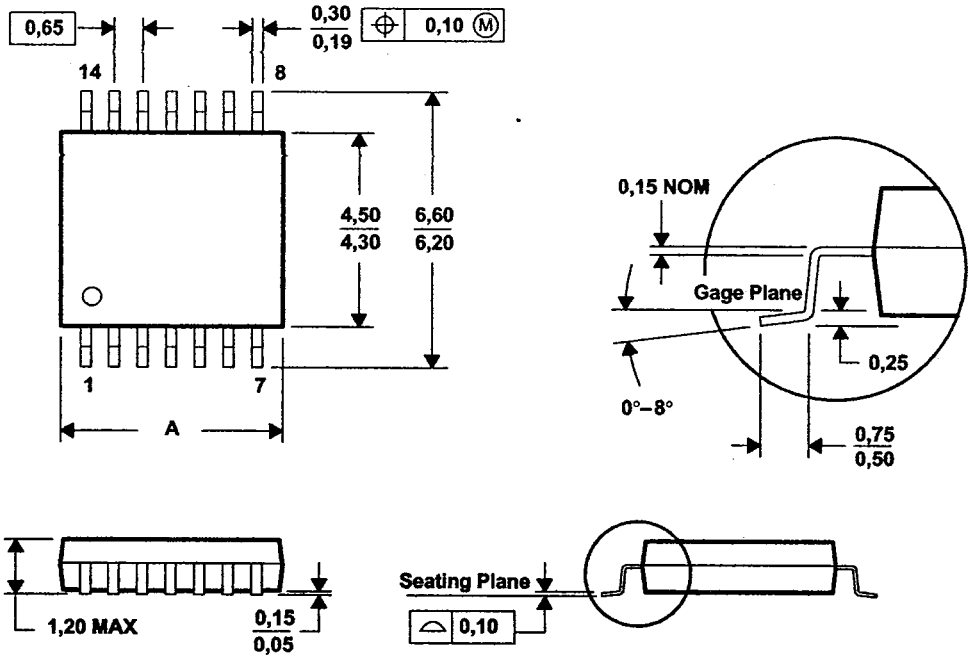


NOTES:
A. All linear dimensions are in millimeters. Dimensioning and tolerancing per ASME Y14.5M–1994.
B. This drawing is subject to change without notice.
C. Quad Flatpack, No–Leads (QFN) package configuration.
D. The package thermal pad must be soldered to the board for thermal and mechanical performance. See the Product Data Sheet for details regarding the exposed thermal pad dimensions.
E. Falls within JEDEC MO–220.

MTSS0019 JANUARY 1995 REVISED FEBRUARY 1999

PW (R-PDSO-G)** **PLASTIC SMALL-OUTLINE PACKAGE**

14 PINS SHOWN



DIM \ PINS **	8	14	16	20	24	28
A MAX	3,10	5,10	5,10	6,60	7,90	9,80
A MIN	2,90	4,90	4,90	6,40	7,70	9,60

4040064/F 01/97

NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion not to exceed 0,15.
D. Falls within JEDEC MO-153

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

Copyright © 2005, Texas Instruments Incorporated

ภาคผนวก ง.
คู่มือการใช้งาน

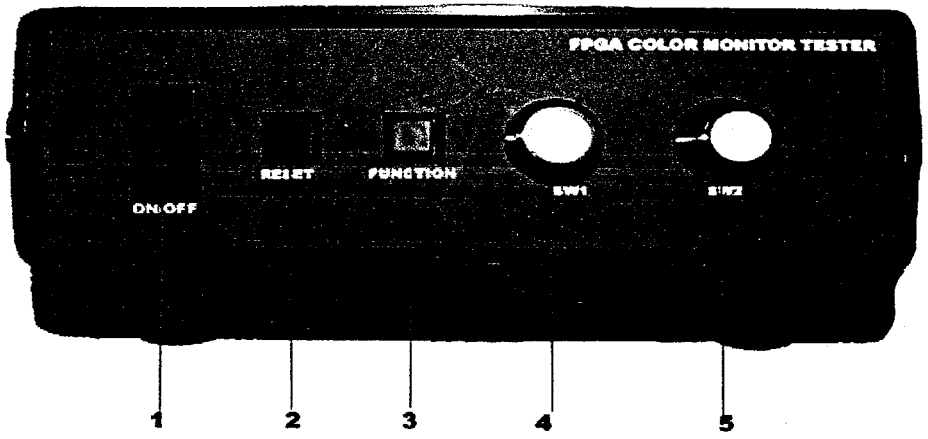
คู่มือการใช้งาน

เครื่อง FPGA COLOR MONITOR TESTER ที่ใช้ตรวจเช็คจอเครื่องนี้เป็นเครื่องทดสอบจอที่นำ FPGA มามีส่วนช่วยในการออกแบบสัญญาณในการทดสอบจอมอนิเตอร์

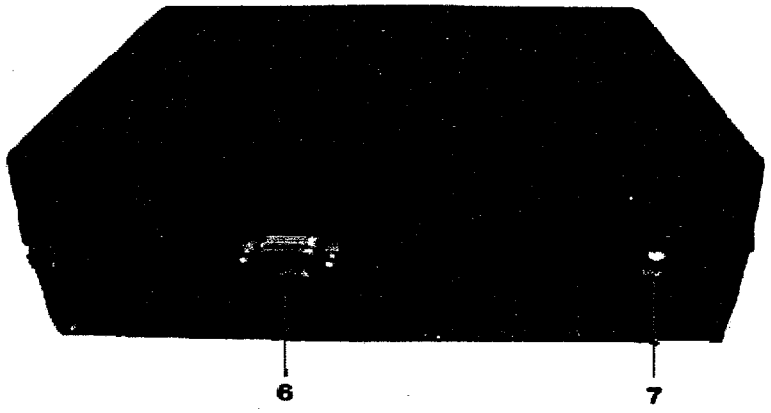
โดยทั่วไปแล้วสัญญาณที่ใช้ทดสอบจอมอนิเตอร์จะเป็นสัญญาณที่ทดสอบการแสดงสีแต่ละสี โดยทำการส่งสัญญาณสีแต่ละสีมาทดสอบที่ละสี แล้วสังเกตผลว่าแต่ละสีทำงานเป็นปกติหรือไม่ ซึ่งถ้ามีความผิดปกติในการแสดงสีจะได้นำไปแก้ไขต่อไป

เพียงเสียบเข้าคอนเน็กเตอร์ของจอมอนิเตอร์ จ่ายไฟเข้าเครื่อง แล้วกดสวิตซ์เท่านั้น ก็สามารถรู้ได้เลยว่าจอที่ท่านกำลังทดสอบอยู่นั้นดีหรือเสียโดยแสดงดังต่อไปนี้

โครงสร้างและส่วนประกอบของเครื่อง



รูปแสดงส่วนประกอบด้านหน้าของเครื่อง



รูปแสดงส่วนประกอบด้านหลังของเครื่อง

ส่วนประกอบต่างๆของเครื่อง FPGA COLOR MONITOR TESTER

1. สวิตซ์ใช้สำหรับปิด – เปิดเครื่อง
2. สวิตซ์รีเซต (RESET) ใช้สำหรับรีเซตฟังก์ชั่นการทำงาน
3. สวิตซ์ฟังก์ชั่น (FUNCTION) ใช้สำหรับเลือกฟังก์ชั่นการแสดงสี
4. ซีเล็คเตอร์สวิตซ์1 (SW1) ใช้สำหรับเลือกการแสดงสี
5. ซีเล็คเตอร์สวิตซ์2 (SW2) ใช้สำหรับเลือกการแสดงสี
6. ช่องเสียบ VGA ของมอนิเตอร์
7. ช่องเสียบไฟ DC 18 V.

แสดงฟังก์ชั่นเลือกการแสดงสี

Mode	Horiz.(kHz)	Vert.(Hz)	Pixel	สวิตซ์เลือกแสดงสี 1 (SW1)				สวิตซ์เลือกแสดงสี 2 (SW2)			
				0	1	2	3	0	1	2	3
0	31.5	60	640x480								
1	31.5	60	640x480								
2	31.5	60	640x480								
3	31.5	60	640x480								
4	31.5	60	640x480								
5	31.5	60	640x480								
6	31.5	60	640x480								
7	31.5	60	640x480								
8	31.5	70	640x350								
9	31.5	70	640x400								
10	35	56	800x600								
11	48	60	1024x768								
12	แสดงฟังก์ชั่นอัตโนมัติฟังก์ฟังตั้งแต่ฟังก์ชั่น 0-7 ที่ SW1 = 0										
13	แสดงฟังก์ชั่นอัตโนมัติฟังก์ฟังตั้งแต่ฟังก์ชั่น 8-11 ที่ SW1 = 0										
14	แสดงฟังก์ชั่นอัตโนมัติฟังก์ฟังตั้งแต่ฟังก์ชั่น 0-11 ที่ SW1 = 0										