สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องตรวจสอบสีจอ VGA แบบ 24 บิต

COLOR TEST ON VGA 24 BIT



โดย

นายศิรพงษ์ สัมฤทธิ์ 43515932

นายสุชาติ เก็งซี 43515938

อาจารย์ที่ปรึกษา

รศ.ดร.มนัส สังวรศิลป์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประจำภาคเรียนที่ 2 ปีการศึกษา 2545

เลขหมู่.....

เลขทะเบียน 50402

วัน,เดือน,ปี 13 พ.ค. 2547

.b.....

i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบฟอร์มรับรองความพร้อมในการสอบ

เครื่องตรวจสอบสีจอ VGA แบบ 24 บิต

COLOR TEST 24 BIT

นายศิรพงษ์ สัมฤทธิ์ รหัส 43515932

นายสุชาติ เก็งซี รหัส 43515938

โครงงานนี้ได้รับการตรวจสอบแล้ว พร้อมที่จะทำการสอบได้

.....

(รศ.ดร.มนัส สังวรศิลป์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องทดสอบสีจอ VGA แบบ 24 บิต

ผู้จัดทำ
นายศิรพงษ์ สัมฤทธ์ 43515932
นายสุชาติ เก็งซี 43515938
อาจารย์ที่ปรึกษา
รศ.ดร.มนัส สังวรศิลป์
ปีการศึกษา 2545

บทคัดย่อ

เครื่องตรวจสอบสีสำหรับจอ VGA ด้วย FPGA (Field Programmable Gate Array) ด้วยความละเอียด 24 บิต ที่นำเสนอในโครงงานนี้เป็นการนำ FPGA มาประยุกต์ใช้งานในการสร้างสัญญาณต่างๆให้กับจอ VGA เพื่อทดสอบสีของจอภาพซึ่งจะทำการออกแบบให้วงจรมีขนาดเล็กลงและไม่สิ้นเปลืองสามารถตรวจสอบสีได้ทุกสีโดยในการออกแบบจะใช้ภาษา VHDL ซึ่งเป็นภาษาในทางดิจิตอลในการเขียนโปรแกรมเพื่อควบคุม FPGA ให้สร้างสัญญาณสี RGB และสัญญาณ sync จากนั้นนำสัญญาณที่ได้ต่อกับพอร์ตของจอ VGA เพื่อแสดงสีต่างทางจอภาพซึ่งสารถเลือกลักษณะของสีที่ตรวจสอบได้โดยทำการเลือกที่สวิทท์จากการทำโครงงานนี้จะทำให้เราได้รับความรู้ในการประยุกต์ใช้งาน FPGA และได้เครื่องตรวจสอบสีของจอ VGA ที่ต้นทุนต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COLOR TEST ON VGA 24 BIT

Mr.Sirapong Sumrit 43515932

Mr.Suchat Kenghi 43515938

Advisor

Assoc.Prof.Dr.Manas Sangworasil

Academic year 2545

ABTRACT

Color test on vga 24 bit by FPGA (Field Programmable Gate Array) present in this project use FPGA produce signal for test color of VGA. For disigning will be small circuit and don't consume. It can test true color. We use VHDL language for design and control FPGA. It is digital language to produce RGB signal and sync signal. We use signal connect with VGA connector port for monitor display. We can select color to test by selectors switch. The benefit from this project is knowledge to FPGA application and save a cost to spending of color test on VGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
Abstract	II
สารบัญ	III
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎี	2
2.1 การกำเนิดการแสดงผลภาพให้กับจอ VGA	2
2.2 เทคโนโลยีในการแสดงผลภาพ	2
2.3 การรีเฟรชของภาพ	5
2.4 คุณสมบัติของแสงและสี	6
2.5 ความสำคัญสามประการของแสงสีที่มองเห็น	6
2.6 การผสมและการแยกแสงสี	6
2.7 องค์ประกอบของภาพ	7
2.8 การสแกนและเรื่องที่เกี่ยวข้อง	8
บทที่ 3 เอฟพีจีเอ	11
3.1 Field Programmable	11
3.2 Mask Programmable	15
3.3 เอฟพีจีเอ (FPGA : Field Programmable Gate Array)	17
บทที่ 4 ภาษา วีเอชดีแอล	22
4.1 ประวัติความเป็นมาของภาษา VHDL	24
4.2 องค์ประกอบพื้นฐานของ VHDL	27
4.3 ภาษาวีเอชดีแอลเพื่อการสังเคราะห์	39
4.4 การบรรยายเชิงพฤติกรรม	40
4.5 การออกแบบจากบนลงล่าง	44
บทที่ 5 การใช้งานโปรแกรม MAX+PLUSE II เบื้องต้น	47
บทที่ 6 การออกแบบและการสร้าง	63
6.1 การออกแบบส่วนฮาร์ดแวร์	63
6.2 การออกแบบส่วนซอร์ฟแวร์	65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันคอมพิวเตอร์ได้เข้ามาเป็นส่วนหนึ่งของชีวิตประจำวันของเราส่วนประกอบสำคัญอย่างหนึ่งคือจอภาพซึ่งเป็นส่วนของการแสดงผลจอภาพที่ดีต้องมีประสิทธิภาพในการแสดงภาพที่ดีมีสีคมชัดดังนั้นจึงจำเป็นต้องมีอุปกรที่เอาไว้ตรวจสอบการแสดงผลโดยทางด้านสีสันของภาพว่ารายละเอียดของภาพถูกต้องและครบสมบูรณ์หรือไม่ ซึ่งวิธีที่จะกล่าวถึงนี้ เป็นการทดสอบด้วย FPGA (Field Programmable Gate Array) ด้วยความละเอียดขนาด 24 บิต โดยการใช้ การเขียนโปรแกรมด้วยภาษา VHDL เพื่อควบคุมการทำงานของ FPGA โดยจอภาพประกอบด้วยแม่สีหลัก คือ สีแดง (red) สีเขียว(green) สีน้ำเงิน(blue) หรือเรียกว่า RGB ซึ่งจะทำการตรวจสอบคือแม่สีหลักรวมไปถึงสีต่างๆที่ได้จากการผสมของแม่สีด้วยวิธีนี้จะชาวยทำให้เราสามารถตรวจสอบคุณภาพทางด้านสีสันของจอภาพคอมพิวเตอร์ได้อย่างถูกต้อง

1.1 วัตถุประสงค์

1.ศึกษาการทำงานและการใช้งานอุปกรณ์ FPGA
2. ศึกษาการใช้ภาษา VHDL เพื่อควบคุม FPGA
3. ศึกษาการเขียนโปรแกรม MAX PLUS II
4. สร้างเครื่องตรวจสอบสีของจอ VGA

1.2 ขอบเขตของโครงงานสร้าง

สร้างเครื่องตรวจสอบสี VGA ที่สามารถตรวจสอบสีของจอคอมพิวเตอร์ได้อย่างถูกต้องและครบสมบูรณ์ โดยเป็นเครื่องที่มีคุณภาพที่ดี มีขนาดเล็กและมีราคาถูก

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถประยุกต์ใช้งานอุปกรณ์ FPGA ได้อย่างกว้างขวาง
2. สามารถเขียนโปรแกรมควบคุมการทำงานของ FPGA ได้
3. ได้เครื่องทดสอบสีของจอ VGA ที่มีคุณภาพดี

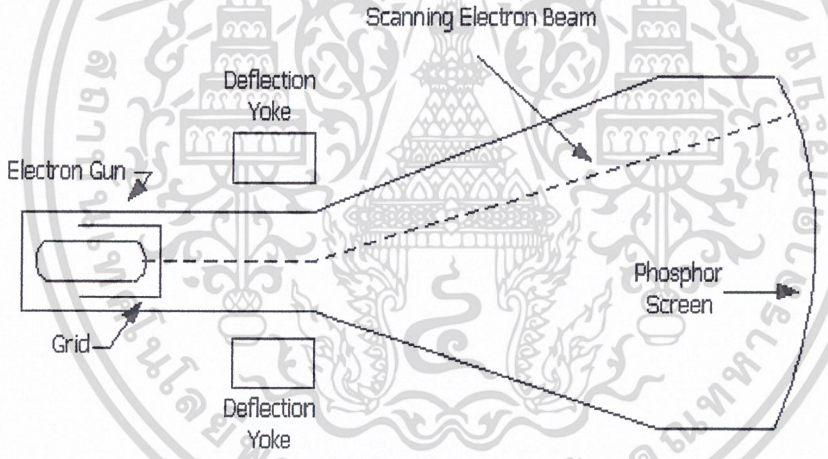
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎี

2.1 การกำเนิดการแสดงผลภาพให้กับจอ VGA (VGA Video Display Generator)

VGA Signal ประกอบด้วย 5 สัญญาณด้วยกัน โดยมี 2 สัญญาณที่เกี่ยวข้องกับระดับสัญญาณของ TTL Logic นั่นคือสัญญาณ Horizontal Sync และ Vertical Sync ซึ่งใช้สำหรับสังเคราะห์สัญญาณภาพ ส่วนอีก 3 สัญญาณจะเกี่ยวข้องกับสัญญาณที่เป็น Analog โดยมีค่า 0.7-1.0 Vp-p จะใช้ในควบคุมสัญญาณสี ซึ่งสัญญาณสีจะประกอบด้วย สีแดง ,สีเขียว ,สีน้ำเงิน สัญญาณสีทั้งหมดจะรวมกันเรียกว่าสัญญาณ RGB

2. 2 เทคโนโลยีในการแสดงผลภาพ (Video Display Technology)

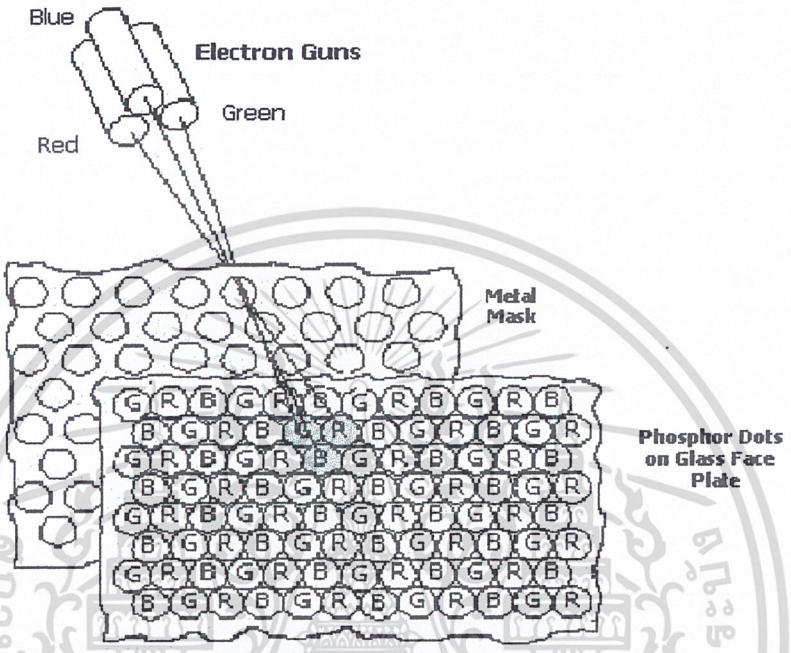


รูปที่ 2.1 แสดงการเบี่ยงเบนลำอิเล็กตรอน ซึ่งควบคุมการทำงานด้วย Yoke

เทคโนโลยีที่ใช้โดยส่งการแสดงผลภาพเป็นสัญญาณภาพนิ่งที่เหมือนธรรมชาติในรูปแบบของสัญญาภาพ ส่วนประกบหลักในจอ VGA ของ Computer จะเป็นจอ CRT สี (Color CRT) หรือหลอดภาพสูญญากาศ “Cathode Ray Tube” (CRT) ลำอิเล็กตรอนจะกวาดสแกนบนม่านฉากตามแนวนอน (Horizontal) ที่ต่อเนื่องกัน เพื่อที่จะกำเนิดเป็นภาพออกมา Yoke สำหรับเบี่ยงเบนลำอิเล็กตรอน (Deflection Yoke) จะทำงานโดยจะอาศัยแม่เหล็ก หรือสนามไฟฟ้าสถิต(electronstatic fields) เบี่ยงเบนลำอิเล็กตรอนให้ไปที่ตำแหน่งที่เหมาะสมบนผิวหน้าของจอภาพ ข้อมูลของสัญญาณ RGB ที่ออกมาเป็นสัญญาณภาพจะใช้ควบคุมความแรงของลำอิเล็กตรอน และแสงที่เกิดขึ้นบนจอภาพนั้นเกิดจาก เมื่อมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

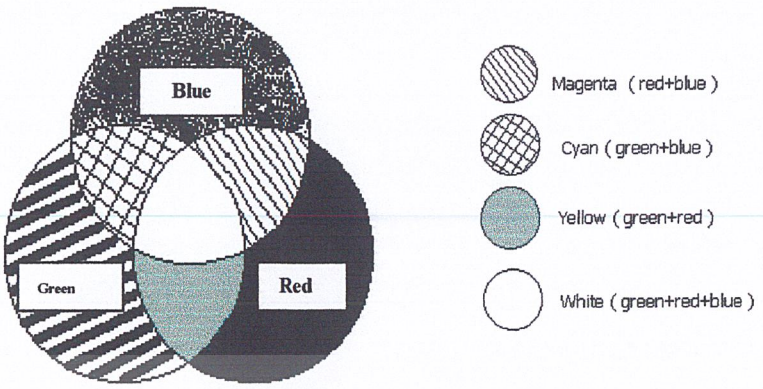
สัญญาณภาพเข้ามาซึ่งจะทำให้ลำอิเล็กตรอนทำงาน และสามารถพุ่งไปปะทะจุดสีเรืองแสงบนผิวหน้าของจอ CRT ผิวหน้าของจอ CRT จะบรรจุไปด้วย สารเรืองแสงที่แตกต่างกัน 3 สี สีหนึ่งสีของสารเรืองแสงจะใช้สำหรับแทนสีของแต่ละแม่สี คือ สีแดง , สีเขียว , สีน้ำเงิน ดังรูปที่ 2.2



รูปที่ 2.2 แสดงการยิงลำอิเล็กตรอนผ่านฉากโลหะไปกระทบจุดเรืองแสงบนผิวจอภาพ

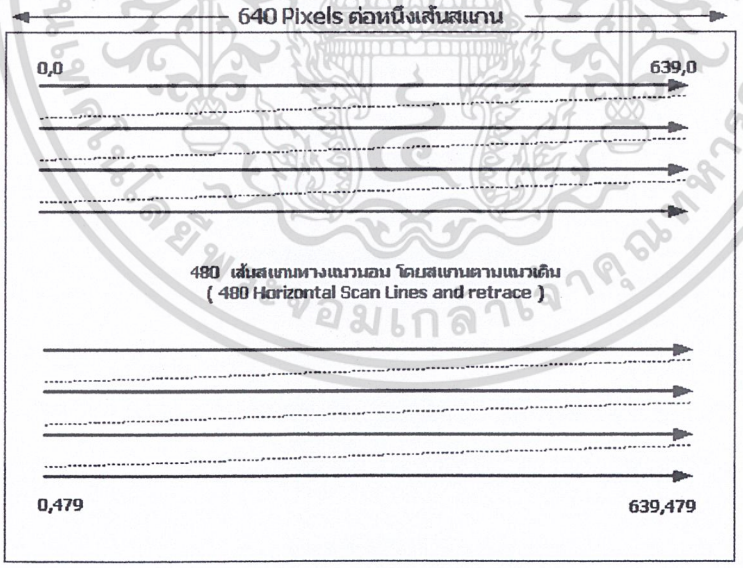
รูปแบบพื้นฐานของ VGA ม่านฉากของจอภาพจะบรรจุไปด้วยจำนวนจุด pixels 640*480 pixels ทั่วทั้งจอภาพ สัญญาณภาพจะต้องทำการ refresh หรือ สแกนใหม่ทั้งจอภาพ 60 ครั้ง/วินาที [60Hz, (60 ครั้ง/sec)] เพื่อที่จะทำให้เกิดการเคลื่อนที่ของภาพนิ่ง และลดการสั่นของภาพ (การกระตุกของภาพ) ซึ่งในหนึ่งรอบของการทำงานนี้เรียกว่า "REFRESH RATE" ในการเกิดการ Refresh นี้จะเป็นการที่ทำให้ข้อความ หรือภาพที่เกิดขึ้นไม่ให้หายไปจากจอภาพ และให้ภาพเกิดขึ้นตลอดเวลา ซึ่งตามนุษย์สามารถพบเห็น การสั่น, การกระพริบที่ REFRESH RATE น้อยกว่า 50 Hz (50 ครั้ง / วินาที)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



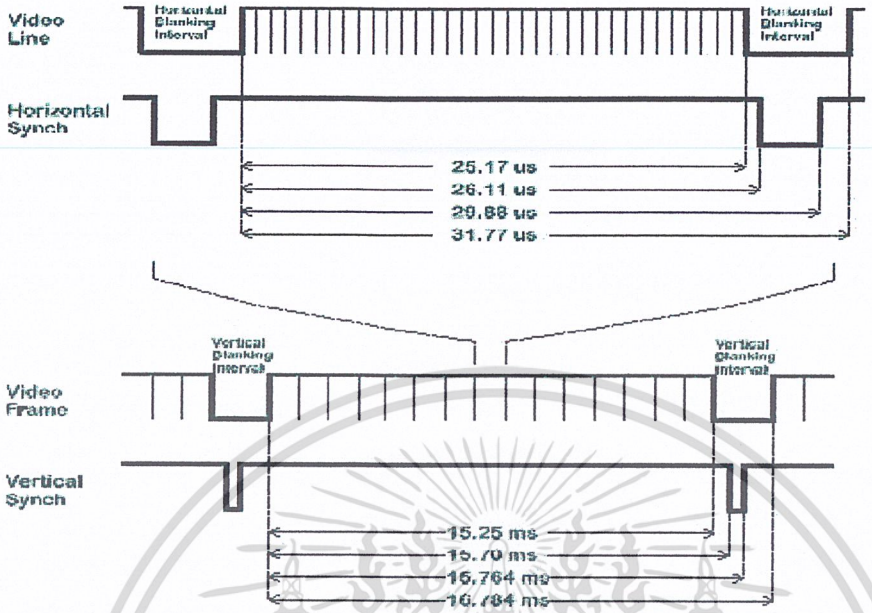
รูปที่ 2.3 แสดงการผสมสีจากแม่สีทั้ง 3 สี โดยที่สีทั้ง 3 ผสมกันแล้วจะได้สีขาว

การลดการสั่น,การกระพริบที่เกิดจากสัญญาณรบกวนจากแหล่งกำเนิดแสงฟลูออเลสเซนต์ ทำได้โดยใช้ REFRESH RATE สูงกว่า 60 Hz เป็นบางทีที่ใช้ในจอภาพของ PC โดยใช้สัญญาณ CLK ควบคุม ส่วนสีของแต่ละ pixel จะกำหนดโดยค่าของสัญญาณ RGB เมื่อสัญญาณสแกนผ่านแต่ละ Pixel ในรูปแบบ 640*480 pixels ด้วย REFRESH RATE 60 Hz จะได้ความเร็วประมาณ 40 nS/pixel โดยจะใช้ CLK ความถี่ 25 MHz ที่มีคาบเวลาแต่ละคาบเวลาเท่ากับ 40 nS



รูปที่ 2.4 แสดงการสแกนทางด้านแนวนอน (Hor.) และแนวตั้ง (Ver.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 (a.) แสดง Timing Diagram ของ Hor. Sync.

(b.) แสดง Timing Diagram ของ Ver. Sync.

2.3 การรีเฟรชของภาพ (Video Refresh)

กระบวนการ Refresh หรือเริ่มสแกนใหม่ (screen refresh) ด้างหน้าจอ จะเริ่มที่ส่วนมุมบนซ้าย และกวาดไป 1 pixel ใน 1 ครั้ง จากซ้ายไปขวา เมื่อสิ้นสุดการสแกนในแถวแรก ก็เริ่มสแกนแถวใหม่ โดยที่ column address จะ reset เริ่มใหม่ที่ column แรก แต่ละแถวจะกวาดจนกระทั่ง pixel ทั้งหมดแสดงผลออกมา และใน 1 ครั้งที่ทั้งหน้าจอมีการกวาดสแกนจนครบ จะทำการเริ่มกระบวนการใหม่ที่จุดเริ่มต้นอีกครั้ง หรือ การ Refresh นั่นเอง

การกวาดสแกนสัญญาณภาพ หรือการเริ่มใหม่ของภาพ จะเป็นไปตามกระบวนการของสัญญาณ Vertical Sync ในรูป 2.5 (b.) ซึ่งบอกถึงการเริ่มแสดงผลของภาพใหม่ หรือ Frame ใหม่ และเริ่มที่จุดสูงสุดทางด้านซ้าย ที่จุด pixel (0,0) และสัญญาณ Horizontal Sync ดังรูป 2.5 (a.) แสดงถึงการเริ่มใหม่ในแถวอื่นๆ หลังจากครบ 480 แถว และ 640 pixels ตามแนวนอนแถวสุดท้าย สัญญาณ Vertical Sync ก็จะทำการ reset จอภาพไปที่มุมซ้ายบนสุดและเริ่มกระบวนการต่อไป ช่วงระหว่างแต่ละครั้งที่ไม่มีข้อมูลแสดงผล และลำอิเล็กตรอนกลับไปอยู่ทางซ้ายของ column เพื่อที่จะเริ่มใหม่ในเส้นสแกนทางด้าน Hor. เส้นอื่น สัญญาณ RGB จะเซ็ตเป็นสีดำ หรือRGB ทั้งหมดเป็น “000” จอภาพ VGA จะดับ ถ้าสัญญาณ Sync ทั้ง 2 นั้นค่าไม่ถูกต้อง จอภาพของ PC ส่วนมาก จะมี LED แสดงสถานะของจอภาพ ถ้า LED เป็นสีเขียว แสดงให้เห็นถึงว่ามีการตรวจพบสัญญาณ Sync และถ้า LED เป็นสีเหลือง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงว่าตรวจไม่พบสัญญาณ Sync ในการ์ดจอภาพของ PC หน่วยความจำในการ์ดจะทำหน้าที่เฉพาะคือใช้เก็บค่าของสีทุกๆ pixels หรือภาพที่ต้องการแสดงผล เวลาแสดงผล หน่วยความจำนี้จะทำการอ่านออกไป เหมือนผลิตสัญญาณ RGB และยิงลำสแกนผ่านฉากของจอภาพออกไป

2.4 คุณสมบัติของแสงและสี

แสงโดยทั่วไปมีอยู่ 2 ประเภท คือแสงที่สายตามนุษย์เรามองเห็น (visible rays) กับแสงที่สายตาคนธรรมดาเรามองไม่เห็น ซึ่งได้แก่แสงจำพวก แกมม่า-เรย์ (gramma-rays), เอ็กซ์-เรย์ (X-rays) อุลตราไวโอเล็ต เรย์ (ultraviolet rays) และ อินฟราเรด เรย์ (infrared rays) แสงเหล่านี้ ต่างก็เป็นส่วนหนึ่งของพลังงานแม่เหล็กไฟฟ้า แต่มีความถี่หรือความยาวคลื่นแตกต่างกันออกไป ส่วนของคลื่นแม่เหล็กไฟฟ้าที่น่าสนใจในเรื่องราวของโทรทัศน์สีนี้ ก็คือส่วนของคลื่นแม่เหล็กไฟฟ้าที่เป็นแสงซึ่งมองเป็นด้วยสายตาของคนธรรมดา ซึ่งมีความยาวคลื่นประมาณ 380 นาโนเมตร จนถึงประมาณ 780 นาโนเมตร แสงที่มองเห็นนี้ จะทำให้สายตาของคนเราได้รับความรู้สึก 2 ประการ ประการแรกคือความรู้สึกว่ามีแสงสว่างมากหรือน้อย (sensation of brightness) ความรู้สึกอีกประการหนึ่ง ก็คือทำให้รู้จักว่าเป็นแสงสีอะไร ความรู้สึกนี้เรียกว่า sensation of color

2.5 ความสำคัญสามประการของแสงสีที่มองเห็น

แสงสีที่มองเห็นจะทำให้เกิดความรู้สึกที่สำคัญอยู่ 3 ประการ คือ เกิดความรู้สึกในเรื่องแสงสี (hue) เกิดความรู้สึกในเรื่องการส่องสว่าง (brightness) และเกิดความรู้สึกในเรื่องแสงสีอิ่มตัว (saturation) โดยความรู้สึกในเรื่องแสงสี (hue) จะทำให้สายตาสามารถแยกแยะออกได้ว่าแสงสีที่มองเห็นเป็นแสงสีแดง หรือแสงสีเขียว หรือแสงสีน้ำเงิน เป็นต้น ส่วนความรู้สึกในเรื่องการส่องสว่าง (brightness or value) จะทำให้สามารถมองเห็นเป็นแสงสีแดงเข้มสว่างหรือมืด เป็นต้น สำหรับความรู้สึกในเรื่องแสงสีอิ่มตัว (saturation or chroma) จะทำให้สามารถรู้ความบริสุทธิ์ของแสงสีได้ว่าเป็นแสงสีน้ำเงินชัดเจน หรือแสงสีน้ำเงินจาง ดังนี้เป็นต้น

2.6 การผสมและการแยกแสงสี

การผสมแสงสีมีลักษณะพิเศษต่างไปจากการผสมสีที่ใช้ในการวาดเขียนภาพสี การผสมสีโดยทั่วไป ซึ่งใช้ในการวาดเขียนหรือการพิมพ์ภาพสี จะทำให้ได้สีผสมที่เข้มข้นกว่า ซึ่งเป็นลักษณะวิธีของ Subtractive mixture ส่วนการผสมแสงสีที่ใช้ในโทรทัศน์สีนั้น จะทำให้ได้แสงสีผสมที่เจือจางกว่าเดิม อันเป็นลักษณะวิธีของ additive mixture การผสมแสงสีที่ใช้ในโทรทัศน์สีนั้นเป็นการผสมแม่สี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(primary color) เพื่อทำให้เกิดแสงสีต่างๆ ขึ้น สีที่เรียกว่าแม่สีนี้จะต้องเป็นอิสระ และจะต้องไม่สามารถนำสีอื่นๆ ที่มีอยู่มาผสมทำให้เกิดแม่สีนั้นๆ ได้เลย แม่สีในเรื่องของโทรทัศน์สีนี้มีอยู่ 3 สีด้วยกัน คือ แสงสีแดง (red) แสงสีเขียว (green) และแสงสีน้ำเงิน (blue)

แสงสีต่างๆ ที่สายตาสามารถมองเห็นนั้น เกิดจากการผสมสีของแม่สีทั้งสามนี้ทั้งนั้น ซึ่งอาจเกิดจากการนำสีใดสีหนึ่งมาผสมกับอีกสีใดสีหนึ่ง หรืออาจเกิดจากการนำสีใดสีหนึ่งมาหักออกจากแสงสีใดสีหนึ่งก็ได้ ยกตัวอย่างเช่น

แสงสีเหลือง (yellow) = แสงสีแดง (red) + แสงสีเขียว (green)

แสงสีเขียวน้ำเงิน (cyan) = แสงสีเขียว (green) + แสงสีน้ำเงิน (blue)

แสงสีม่วง (magenta) = แสงสีน้ำเงิน (blue) + แสงสีแดง (red)

แสงสีแดง + แสงสีเขียว + แสงสีน้ำเงิน = แสงสีขาวหรือไม่มีสี (white)

แสงสีจากแม่สีทั้งสาม คือ แสงสีแดง แสงสีเขียว และแสงสีน้ำเงินนี้ ได้มีการทดลองหาความยาวคลื่นที่แน่นอนไว้แล้ว คณะกรรมการระหว่างประเทศในเรื่องการส่องสว่าง (International Committee on Illumination หรือ Commission International de I'Eclairage หรือเรียกชื่อสั้นๆ ว่า CIE) ได้กำหนดความยาวคลื่นของแสงจากแม่สีทั้งสามไว้ในตาราง

ตารางแสดงแม่สีทั้งสามในระบบแสงสีแดงแสงสีเขียวแสงสีน้ำเงิน (RGB)

รายการ	ความยาวคลื่น (นาโนเมตร)	หมายเหตุ
สีน้ำเงิน	435.8	เป็นส่วนหนึ่งของสเปคตรัมเชิงเส้นของปรอท
สีเขียว	546.1	เป็นส่วนหนึ่งของสเปคตรัมเชิงเส้นของปรอท
สีแดง	700	ประมาณค่าออกเป็นตัวเลขหยาบๆ

2.7 องค์ประกอบของภาพ

หากเราตัดภาพจากหนังสือพิมพ์สักภาพหนึ่ง แล้วขยายขึ้นด้วยกล้องหรือแว่นขยาย จะพบว่าภาพมีองค์ประกอบมาจากจุดสีขาวและจุดสีดำมากมาย มาเรียงกันประกอบขึ้นเป็นภาพ จุดเหล่านี้เองที่เรียกว่าองค์ประกอบของภาพ หรือพิคเจอร์ อีลีเมนต์ (Picture Element) หรือพิกเซล (Pixel)

ทำนองเดียวกัน ภาพที่ปรากฏทางจอภาพก็เอามาจากหลักการนี้ ภาพที่เกิดขึ้นบนจอภาพหรือจอโทรทัศน์ประกอบด้วยเส้นขวางเล็กๆ ในแนวนอนเป็นจำนวนมาก แต่ละเส้นนั้นมีทั้งส่วนที่ดำสนิท ส่วนที่จาง และส่วนที่สว่างรวมกันอยู่ เส้นเหล่านี้เราได้มาจากการกวาดลำแสง (Scan) ความแตกต่างกันบนเส้นกวาดลำแสงหรือเส้นสแกนเหล่านี้เองที่เราจัดว่าเป็นองค์ประกอบภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบสแกน 525 เส้นเราจะแบ่งส่วนทางแนวตั้งได้ 700 เส้น ดังนั้นจึงกล่าวได้ว่าหากจะหาจำนวนขององค์ประกอบภาพในระบบการสแกน 525 เส้นแล้ว จะได้เท่ากับ 525×700 เท่ากับ 367,500 พิกเซล ยิ่งภาพมีจำนวนเส้นมากเท่าไรรายละเอียดภาพยิ่งมากขึ้นเท่านั้น จอภาพที่มีจำนวนเส้นสแกนมากย่อยได้รายละเอียดหรือความชัดเจนของภาพมากกว่า แต่การออกแบบวงจรจะยากตามไปด้วย เนื่องจากแบนด์วิดธ์ของความถี่จะต้องกว้างขึ้นด้วย จากที่กล่าวมาว่าองค์ประกอบของภาพมีหลายพิกเซล แต่ในความเป็นจริงแล้วเรามองเห็นได้ไม่ครบทุกพิกเซล เนื่องจากมีบางส่วนหายไประหว่างการสแกน จากการทดลองพบว่าภาพที่พอดูได้จะมีองค์ประกอบไม่ต่ำกว่า 200,000 พิกเซล

ระบบที่ใช้อยู่ในประเทศไทย ปัจจุบันเราใช้ระบบการสแกน 625 เส้น 25 ภาพต่อวินาที จึงทำให้แบนด์วิดธ์ต้องกว้างถึง 7 เมกะเฮิรตซ์ ในขณะที่ระบบ 525 เส้น กว้างเพียง 6 เมกะเฮิรตซ์เท่านั้น อย่างไรก็ตามองค์ประกอบภาพจะมีความละเอียดมากขึ้น โดยสามารถหาองค์ประกอบภาพได้จากค่าจำนวนสแกน 625เส้นคูณกับจำนวนจุดหรือองค์ประกอบตามแนวตั้ง851เส้นได้เท่ากับ 531,875 พิกเซล

ปัจจุบันส่วนที่เรียกว่าองค์ประกอบภาพได้ถูกนำเอาไปใช้งานอย่างเป็นจริงมากขึ้น ในโทรทัศน์หรือเครื่องเล่นวีดีโอคาสเซ็ทเร็คคอร์เดอร์ จะมีการนำเอาพิกเซลเหล่านี้เก็บไว้ในหน่วยความจำ เพราะพิกเซลเท่านั้นที่ระบบดิจิตอลจะจัดการข้อมูลได้ เราจะพบวิธีการนี้ในโทรทัศน์ระบบดิจิตอล, โทรทัศน์ระบบคอมพิวเตอร์, โทรทัศน์จอภาพแอลซีดี, ดิจิตอล วีซีอาร์, โทรทัศน์หรือวีซีอาร์ ระบบภาพซ้อนภาพ ฯลฯ

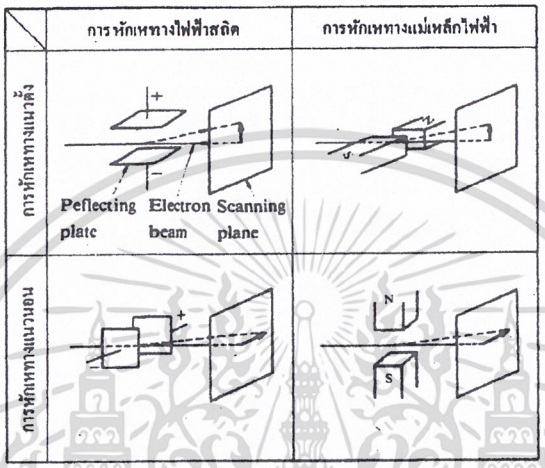
ในปัจจุบันสำหรับโทรทัศน์ธรรมดาเราจะพบว่ามีการเพิ่มเส้นภาพให้มากขึ้น และนั่นแน่นอนว่าจำนวนพิกเซลย่อมมากขึ้นด้วย อย่างโทรทัศน์จอใหญ่หรือโทรทัศน์ที่ต้องการรายละเอียดสูงอย่าง HDTV อาจจะต้องใช้เส้นสแกนภาพมากกว่า 625 เส้น เช่นที่นิยมใช้กันในปัจจุบันคือ 725 เส้น หรืออย่างเครื่องฉายวีดีโอโปรเจ็คเตอร์จะใช้เส้นภาพ 2,200 เส้นภาพ และหากเป็นจอใหญ่หลายร้อยนิ้วจะต้องเพิ่มรายละเอียดมากขึ้นอีก นั่นคือการเพิ่มพิกเซลอีลีเมนต์นั่นเอง

2.8 การสแกนและเรื่องที่เกี่ยวข้อง

ภาพบนจอหลอดภาพของเครื่องรับโทรทัศน์สีโดยทั่วไป จะประกอบด้วยเส้นขวางเล็กๆ ในแนวนอนเป็นจำนวนมาก ซึ่งแต่ละเส้นเหล่านี้ มีทั้งส่วนที่ดำสนิทหรือมีสีเข้ม ส่วนที่ดำจางหรือมีสีจาง และส่วนที่สว่างมากปะปนกันอยู่ เส้นขวางเล็กๆ ในแนวนอนเหล่านี้ มีชื่อเรียกว่า เส้นสแกน เส้นเหล่านี้ประกอบไปด้วยจุดเล็กๆ ซึ่งมีทั้งมืดและสว่างปะปนกันอยู่ ภาพที่ปรากฏบนจอหลอดภาพจึงประกอบด้วยจุดเล็กๆ ที่มีระดับของความสว่างแตกต่างกันเป็นจำนวนมาก จุดเล็กๆ เหล่านี้เรียกว่าส่วนประกอบของภาพ หรือ picture element ซึ่งมีส่วนสัมพันธ์กับความละเอียดของภาพเช่นเดียวกับจุดดำหรือจุดสีเล็กๆ ในรูปภาพของสิ่งตีพิมพ์ ภาพที่เห็นบนจอหลอดภาพจะมองดูละเอียดน่าดู หากจำนวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดเล็กๆ หรือจำนวนเส้นสแกนในแนวนอนมากเพียงพอ อย่างไรก็ตาม ภาพที่เห็นบนจอหลอดภาพจะมองละเอียดหรือหยาบไม่น่าดูอย่างไรนั้น ยังขึ้นอยู่กับส่วนประกอบอีกหลายอย่าง เช่น ความสว่างของภาพและระยะทางที่มองดูภาพ เป็นต้น สำหรับโทรทัศน์ระบบอเมริกัน ซึ่งมีจำนวนเส้นสแกนน้อยกว่า แต่ถ้าหากมองดูภาพในระยะห่างประมาณ 4-8 เท่าของความสูงของภาพแล้ว ก็จะรู้สึกว่าเป็นภาพพอใช้ได้เหมือนกัน

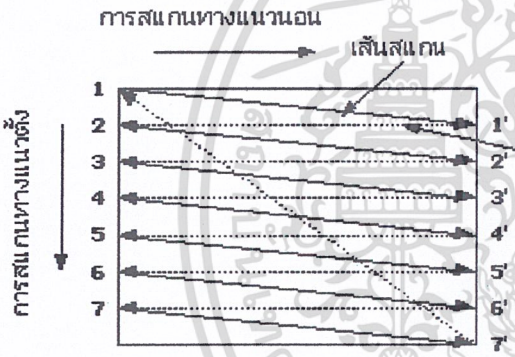


รูปที่ 2.6 ทฤษฎีของการหักเหทางไฟฟ้าสถิต และทางแม่เหล็กไฟฟ้า

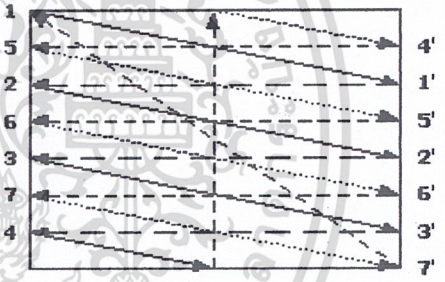
จุดที่เห็นสว่างในจอหลอดภาพของจอภาพ เกิดขึ้นเพราะอิเล็กตรอนที่หลุดออกไปจากแคโทดและถูกดึงดูดให้วิ่งเป็นลำไปกระทบแอโนดหรือจอหลอดภาพ ซึ่งฉาบวัสดุเรืองแสงบางชนิดเอาไว้ จุดที่มีการกระทบกัน ก็จะมองเห็นเป็นจุดสว่างขึ้นที่จอ การสแกนก็คือ การทำให้จุดสว่างนี้เคลื่อนที่ไปในจังหวะที่ต้องการ ซึ่งในเรื่องของโทรทัศน์ ก็ต้องการให้จุดสว่างนี้เคลื่อนที่ไปในแนวนอนและแนวตั้ง โดยอาศัยความเข้มของสนามแม่เหล็กเข้าช่วยเหลือ ทำให้เกิดการดึงดูดหรือการผลักกันกับอิเล็กตรอน ในหลักการ การทำให้เกิดการดึงดูดหรือการผลักกันกับอิเล็กตรอนนี้ อาจทำได้โดยวิธีการหักเหของไฟฟ้าสถิต (electrostatic deflection) หรือวิธีการหักเหของแม่เหล็กไฟฟ้า (electromagnetic deflection) ตามที่แสดงไว้ในรูปที่ 2.6 ซึ่งวิธีการหลังนี้เป็นที่นิยมกันมากในทางปฏิบัติ สนามแม่เหล็กนี้เกิดขึ้นโดยการปล่อยกระแสไฟฟ้ารูปฟันเลื่อยให้ไหลผ่านขดลวดของการหักเห (deflection coil) ที่พันอยู่รอบๆ คอหลอดภาพ ซึ่งมีอยู่ด้วยกันสองชุดคือ ขดลวดที่พันอยู่รอบคอหลอดภาพในแนวนอนชุดหนึ่ง และขดลวดที่พันอยู่รอบคอหลอดภาพในแนวตั้งอีกชุดหนึ่ง สำหรับโทรทัศน์ระบบยุโรป ความถี่ของกระแสรูปฟันเลื่อยที่ไหลผ่านขดลวดของการหักเหในแนวนอน จะมีค่า 16,625 เฮิร์ทซ์ ส่วนกระแสรูปฟันเลื่อยที่ไหลผ่านขดลวดของการหักเหในแนวตั้ง จะมีค่าเพียง 50 เฮิรทซ์ เท่านั้น โดยปกติการสแกนจะเริ่มต้นขึ้นโดยการทำให้จุดสว่างบนจอหลอดภาพเคลื่อนที่จากซ้ายมือด้านบนของจอไปขวามือในแนวนอน ซึ่งเมื่อถึงตำแหน่งขวามือสุด ก็จะถูกเบนต่ำลงเล็กน้อย อันเป็นผลจากการที่มี

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระแสรูปฟันเลื่อยไหลผ่านขดลวดของการหักเหในแนวตั้ง แล้วก็จะกลับไปตั้งต้นใหม่ทางซ้ายมือเพื่อเคลื่อนที่มาทางขวามือในแนวนอนอีก เป็นอยู่เช่นนี้เรื่อยๆ จนกระทั่งจุดสว่างนั้นไปถึงตำแหน่งขวามือข้างล่างสุดของจอหลอดภาพ จึงเป็นอันเสร็จสิ้นการสแกนภาพนิ่งภาพหนึ่ง หรือเรียกกันว่า เฟรมหนึ่ง ตามที่แสดงไว้ในรูป 2.7 หลังจากนั้น ลำอิเล็กตรอนก็จะกลับไปตั้งต้นใหม่ทางซ้ายมือด้านบนสุดของจอหลอดภาพอีกเพื่อสแกนภาพอันดับถัดไป อย่างไรก็ตามเพื่อลดอาการกระพริบของการสแกนภาพนิ่งแต่ละภาพ มักนิยมจัดทำสองครั้งในแบบของการสแกนไขว้กัน ซึ่งเรียกว่า interlace scanning ตามที่แสดงไว้ในรูปที่ 2.8 โดยกำหนดให้ภาพนิ่งหนึ่งเฟรม (frame) ประกอบด้วยภาพนิ่งสองฟิลด์ (field) และเริ่มต้นด้วยการสแกนภาพนิ่งฟิลด์เส้นคี่ก่อน เมื่อเสร็จสิ้นถึงตำแหน่งขวามือล่างสุดของจอหลอดภาพแล้ว จึงกลับไปตั้งต้นใหม่ทางซ้ายมือด้านบนสุด แล้วเริ่มสแกนภาพนิ่งฟิลด์เส้นคู่ต่อไป จนถึงตำแหน่งขวามือล่างสุด หลังจากนั้นจึงจะเริ่มต้นสแกนภาพนิ่งอันดับอื่นต่อไปใหม่ ดังนั้นภาพนิ่งหนึ่งภาพหรือภาพนิ่งหนึ่งเฟรม จึงประกอบด้วยฟิลด์เส้นสแกนเส้นคี่และฟิลด์เส้นสแกนเส้นคู่



รูปที่ 2.7 การสแกนจากซ้ายไปขวาและจากบนลงล่าง

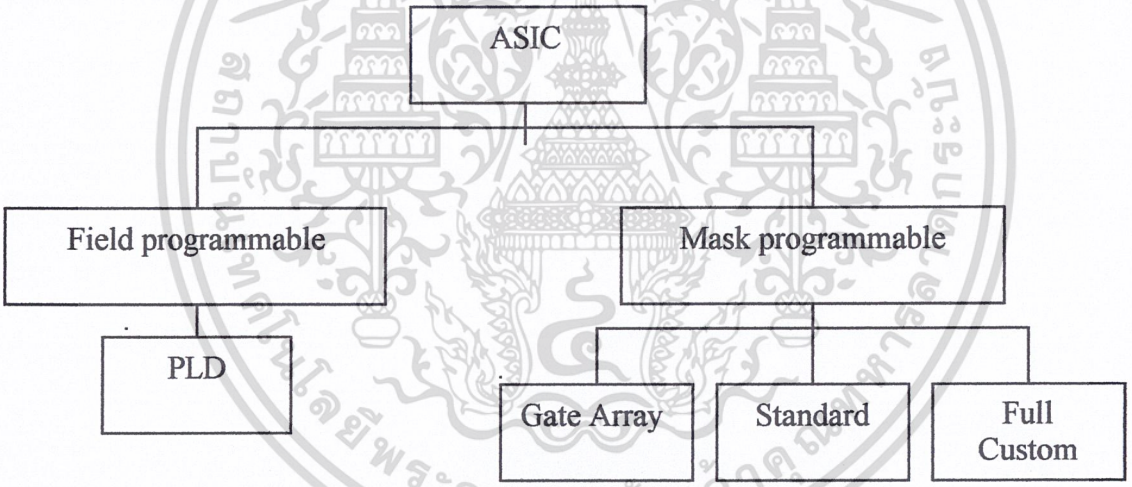


รูปที่ 2.8 การสแกนไขว้กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
เอฟพีจีเอ

ความก้าวหน้าของอุตสาหกรรมอิเล็กทรอนิกส์ปัจจุบันทำให้เกิดการพัฒนาความสามารถของอุปกรณ์ต่างๆ มากมายซึ่งทำให้เกิดการลดค่าใช้จ่าย การสิ้นเปลืองพลังงานและขนาด ในขณะเดียวกันก็มีการเพิ่มประสิทธิภาพและระดับความเชื่อถือได้ของวงจรรวมที่สูงขึ้นเห็นได้ชัดจากเทคโนโลยีไมโครโพรเซสเซอร์และหน่วยความจำปัจจุบัน ทุกๆ ครั้งที่มีการพัฒนาขึ้นทำให้เกิดช่องว่างวงจรรวมและไอซีมาตรฐานมากขึ้น ในการพัฒนาเพิ่มความหนาแน่นและจำนวน ฟังก์ชั่นลอจิกที่เหมาะสม นักออกแบบอุปกรณ์ทางด้านดิจิตอลได้พิจารณาถึงการผลิตให้ขนาดมากๆ และการผลิตวงจรรวม (ASIC: Application Specific Integrated Circuit) ซึ่งวงจรรวมจะแบ่งตามการสร้างออกเป็น 2 กลุ่ม คือ Field programmable และ Mask programmable ดังแสดงในรูปที่ 2.1



รูปที่ 3.1 แสดงผังแสดงการแบ่งกลุ่มของวงจรรวม ASIC

3.1 Field Programmable

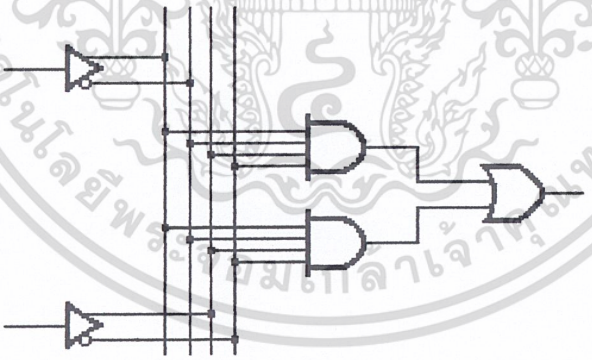
อุปกรณ์วงจรรวมเฉพาะงาน ASIC แบบ field programmable มีอยู่มากมายหลายชนิด แต่มีลักษณะการสร้างหรือกำหนดการทำงานของวงจรที่เหมือนกัน กล่าวคือ ผู้ใช้งานสามารถออกแบบและสร้างวงจรที่ต้องการใช้ลงในตัวอุปกรณ์ได้เองโดยไม่ต้องไปโรงงานเพื่อผลิต โดยเฉพาะอย่างยิ่งในปัจจุบันนี้มีเครื่องมือที่ใช้ช่วยในการออกแบบ และสร้างวงจรร่วมกับไมโครคอมพิวเตอร์ที่มีความสามารถสูงในการพัฒนาตั้งแต่ขั้นการออกแบบ การจำลองการทำงาน จนถึงจัดสร้างวงจรลงในอุปกรณ์ รวมทั้งอุปกรณ์ Field Programmable เหล่านี้สามารถหาซื้อได้ง่ายทำให้การสร้างวงจรอิเล็กทรอนิกส์จน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถึงระบบไมโครโพรเซสเซอร์หันมาใช้อุปกรณ์จำพวกนี้ เป็นอุปกรณ์ประกอบในวงจรแทนอุปกรณ์ย่อยๆ แยกชิ้น (Discrete component)

3.1.1 พีแอลดี (PLD: Programmable Logic Device)

ภายในอุปกรณ์พีแอลดีถูกเตรียมเป็นวงจรพื้นฐานทางด้านลอจิกต่อกันอยู่เป็นกลุ่มมีทั้งวงจรคอมบิเนชั่น (Combination) และซีเควลเชียล (Sequential) ซึ่งมีส่วนประกอบเป็นวงจรภายในเทคโนโลยีของวงจรที่ใช้สร้างพีแอลดีมีทั้ง ทีทีแอล (TTL) อีซีแอล (ECL) และ ซีมอส (CMOS) ตามความเหมาะสมของแต่ละระบบ อุปกรณ์พีแอลดีทุกชนิดมีหลักการพื้นฐานของวงจรภายในที่เหมือนกันโดยมีวงจรหลักเป็นวงจรคอมบิเนชั่นที่ให้ผลเป็นผลคูณร่วมบวก (Sum of product) ประกอบไปด้วยชุดของแอนด์เกตที่ต่อร่วมกับออเกตการโปรแกรมคือ การเลือกว่าจะจะให้มีการต่ออินพุตภายในของแอนด์เกตกับสัญญาณอินพุตใดบ้างซึ่งมีทั้งจากภายนอกและสัญญาณป้อนกลับจากเอาต์พุตภายในเอง การติดต่ออินพุตของออเกตกับเอาต์พุตของ แอนด์เกต ตัวต่างๆ วิธีการเลือกหรือการโปรแกรมทางกายภาพ อินพุตต่างๆ ของอุปกรณ์ทุกตัวจะถูกต่อผ่านฟิวส์เข้ากับแหล่งสัญญาณ ซึ่งถ้าไม่ต้องการใช้สัญญาณใดจะตัดฟิวส์ทำให้สามารถโปรแกรมได้ครั้งเดียว อุปกรณ์พีแอลดีบางชนิดใช้มอสทรานซิสเตอร์แทนฟิวส์ทำให้สามารถโปรแกรมโดยใช้กระแสไฟฟ้า และสามารถลบและโปรแกรมใหม่เข้าไปได้อีก



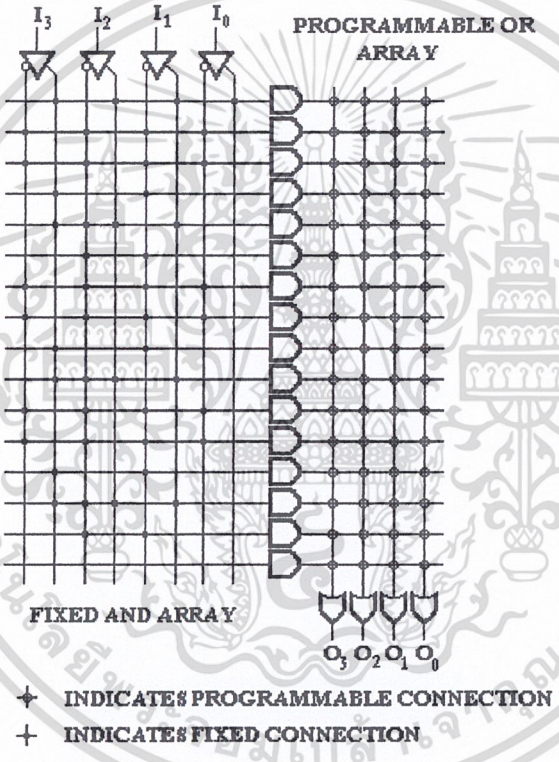
รูปที่ 3.2 แสดงวงจรพื้นฐานของอุปกรณ์พีแอลดีซึ่งอยู่ในรูปผลคูณร่วมบวก

3.1.2 พรอม (PROM: Programmable Read Only Memory)

พรอมคือหน่วยความจำรอม (ROM) ที่โปรแกรมได้ ซึ่งนับว่าเป็นอุปกรณ์พีแอลดี ชนิดหนึ่งซึ่งวงจรภายในของพรอมเสมือนกับประกอบไปด้วยแถวลำดับของแอนด์เกตและออเกต (Amd-Or Array) ผลเอาต์พุตที่ขาต้าเอาต์พุตสามารถแสดงในสมการของฟังก์ชั่นผลคูณร่วมบวก (Sum of product) ของสัญญาณอินพุตที่ขาแอดเดรส รูปที่ 2.3 แสดงถึงลักษณะการต่อเป็นแถวลำดับของแอนด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกตและออเกตของพรอมขนาด 16x4 บิต วงจรทางด้านซ้ายบนสุดเป็น แอนด์เกตที่ให้ผลเป็นผลคูณ (Product) ของกรณีอินพุตเป็น 0000 แอนด์เกตที่อยู่ถัดลงมาเป็นผลคูณของกรณีที่อินพุต เป็น 0001, 0010, ... จนถึงตัวล่างสุดคือผลคูณในกรณีที่อินพุตเป็น 1111 ที่อินพุตแต่ละบิตของหน่วยความจำสามารถเลือกได้ว่าจะให้เป็น 1 ในกรณีที่อินพุตจากแอดเดรส เป็นอย่างไรบ้างเหมือนกันเป็นการนำเอาต์พุตจากผลคูณที่ต้องการให้เอาต์พุตแต่ละบิตเป็น 1 ไปออกันจึงเปรียบเหมือนกับว่าในพรอมมีจำนวนแอนด์เกตเท่ากับจำนวนตำแหน่งความจำและมีออเกตจำนวนเท่ากับจำนวนบิตของสัญญาณข้อมูลออก(Data output) อินพุต่ของออเกตทุกตัวสามารถต่อเข้ากับแอนด์เกตตัวใดก็ได้ทุกตัว ซึ่งอาจเรียกได้ว่าเป็นพีแอลดีแบบ fixed AND/programmable OR



รูปที่ 3.3 แสดงลักษณะของพรอมเมื่อเปรียบเทียบเป็นวงจรในรูปผลคูณร่วมบวก

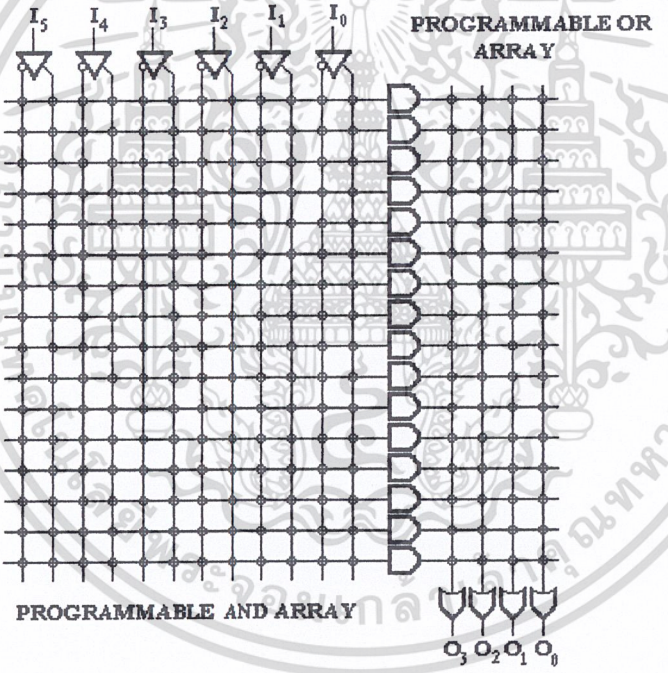
3.1.3 พีเอแอล (PAL: Programmable Array Logic)

ในช่วงกลางปี ค. ศ. 1970บริษัทเอ็มเอ็มไอ (MMI: Monolithic Memory) ในประเทศสหรัฐเมริกา ได้พัฒนาอุปกรณ์พีเอแอล เป็นพีแอลดีชนิดใหม่โดยใช้เทคโนโลยีแบบแอลเอสไอ (LSI: Large Scale Integration) สามารถโปรแกรมเลือกวงจรภายใน โดยใช้ฟิวส์ที่เชื่อมต่ออยู่ระหว่างสัญญาณอินพุตภายนอกและการป้อนกลับจากภายในกับแอนด์เกตที่ต่อเป็นฟังก์ชั่นผลคูณ (Product) อยู่ในตัววงจรรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พีแอลเอ (PLA: Programmable Logic Array)

อุปกรณ์ที่สามารถโปรแกรมได้แบบพีแอลเอเกิดขึ้นเมื่อปี ค.ศ. 1975 โดยบริษัทซิกเนทริกส์ (Signetics)สหรัฐเอมริกา ซึ่งเป็นบริษัทผู้ผลิตวงจรรวมรายใหญ่รายหนึ่ง ผลิตและนำเสนออุปกรณ์โดยใช้ชื่อว่า เอฟพีแอลเอ (FPLA : Field Programmable Logic Array) สามารถโปรแกรมการต่อลอจิกทั้งทางด้านแอนด์เกตและออเกตได้ และยังเลือกเอาต์พุตเป็น active high หรือ active low โดยต่อผ่านเอ็กคูลสีบออเกต ให้ทำหน้าที่เป็นนอนอินเวอร์เตอร์หรือเป็น อินเวอร์เตอร์แล้วแต่ภายในของพีแอลเอต่อมาปี ค.ศ. 1979 บริษัทซิกเนทริกส์ ได้สร้างเอฟพีแอลเอใหม่ที่มีรีจิสเตอร์ต่ออยู่ภายในวงจรเพิ่มขึ้นรวมทั้งสามารถเลือกสัญญาณอินพุตที่มาจากการป้อนกลับจากรีจิสเตอร์ได้ด้วย ทำให้สามารถใช้อุปกรณ์พีแอลเอใหม่นี้สร้างวงจร State machine ได้ อุปกรณ์ใหม่ที่มีรีจิกเตอร์ อยู่ด้วยนี้ถูกตั้งชื่อใหม่เป็น เอฟพีแอลเอส (FPLS: Field Programmable Logic Sequencer) มีทั้งที่เป็นทีทีแอลและซีมอส



รูปที่ 3.4 แสดงวงจรพื้นฐานภายในของพีแอลเอ

แอลซีเอ (LCA: Logic Cell Array)

อุปกรณ์ชนิดนี้ถูกสร้างขึ้นเมื่อประมาณปี ค.ศ. 1986 โดยบริษัทไซริ้ง (XILINX Inc.) ซึ่งเป็นบริษัทที่ร่วมทำการค้นคว้ากับบริษัทเอ็มเอ็มไอ (MMI) สร้างเป็นอาเรย์ที่ประกอบด้วยเกตจำนวน 1,200-1,800 เกต มีลักษณะของสถาปัตยกรรมที่ใกล้เคียงกับเกตอาเรย์ (Gate array) โดยโปรเซสแบบซีมอส 1.6 ไมครอนขึ้นโลหะคู่ (CMOS 1.6 microns double – layer metal) สามารถโปรแกรมและลบได้โดยใช้กระแสไฟฟ้า (Static RAM based) ภายในจัดเรียงเป็นเมทริกซ์ของลอจิกเซลล์ ล้อมรอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายนอกด้วยอินพุต เอาต์พุตเซลล์ อุปกรณ์แอลซีเอตัวแรกของบริษัทไซริ้ง คือ แอลซีเอเบอร์ 2064 ประกอบด้วยเซลล์เรียงเป็นเมทริกซ์มีจำนวน 64 เซลล์ แต่ละเซลล์เรียกว่า ซีแอลบี (CLB : Configurable Logic Block) แต่ในปัจจุบัน ได้ดพัฒนามาอยู่ในรูปของ เอฟพีจีเอ (FPGA : Field Programmable Gate Array) ซึ่งมีประสิทธิภาพ ความจุของเกตสูงมากขึ้น โดยสร้างออกมาเป็นอนุกรม (Series) ต่างๆ เช่น ตระกูล XC 3000 และ ตระกูล XC 4000 เป็นต้น

อีพีแอลดี (EPLD: Erasable P5rogrammable Logic Device)

อีพีแอลดีเป็นอุปกรณ์สามารถโปรแกรมได้ที่สามารถลบและกระทำการโปรแกรมใหม่ได้ เพื่อใช้ทำวงจรต้นแบบ ตัวอย่างได้แก่ พีแอลดี ในอนุกรมอีพี (EP series) ของบริษัทอัลเทอร่า (Altera Ine.) ประเทศสหรัฐเอมริกาซึ่งเป็นบริษัทที่ผลิตอีพีแอลดี เป็นรายแรกโดยเริ่มเมื่อปี ค.ศ. 1984 เป็น พีแอลดี ที่ใช้โปรเซสเหมือนกับซีมอสอี พรอม (CMOS EPROM) คือ ใช้มอสทรานซิสเตอร์เชื่อมต่อระหว่างสัญญาณอินพุตกับจุดที่ต้องการแทนการใช้ฟิวส์ดังเดิม ทำให้สามารถโปรแกรมการต่อวงจรภายในอุปกรณ์ด้วยการจ่ายไฟฟ้าตามขนาดที่กำหนดเข้าไปยังตัวอุปกรณ์ และลบได้โดยใช้แสงอุลตร้าไวโอเลทฉายผ่านช่องหน้าต่างกระจกเข้าไปตกกระทบในตัวซิปของอุปกรณ์

3.2 Mask programmable

การใช้งานวงจรรวม ASIC ในเชิงพาณิชย์ จำเป็นต้องใช้วงจรรวม ASIC แบบ Mask programmable เนื่องจากต้นทุนต่อหนึ่งตัวต่ำกว่าวงจรแบบ Field programmable ASIC ในกรณีที่ปริมาณการผลิตสูงนับพันพันนับหมื่นตัวขึ้นไป ตัวอย่าง เช่น วงจรอีพีแอลดี ตัวหนึ่งอาจสูงถึงหนึ่งพันบาท ในขณะที่ถ้าผลิตวงจรรวมที่มีคุณสมบัติเหมือนกันทุกประการโดยใช้ Mask programmable แล้ว ราคาตัวหนึ่งจะลดลงเหลือเพียงไม่ถึงหนึ่งร้อยบาท การใช้วงจรรวมแบบ Mask programmable จึงมีบทบาทสำคัญในการผลิตสินค้าอิเล็กทรอนิกส์ในเชิงพาณิชย์ในปัจจุบัน

วงจรรวมประเภทนี้ หลังจากผู้ใช้ออกแบบวงจรและตรวจสอบการทำงานจนเป็นที่น่าพอใจแล้ว ต้องส่งให้ผู้ผลิตทำการเจือสาร ไม่สามารถโปรแกรมได้ด้วยตนเองเหมือนกับวงจรรวมแบบ Field programmable ช่วงเวลาการผลิตออกใช้งานจึงใช้เวลานับเดือนและมีค่าใช้จ่ายเบื้องต้นในการเจือสารสูง วงจรรวมแบบ Mask programmable ASIC ในปัจจุบันได้แก่ เกตอาเรย์, เซลล์มาตรฐาน และฟู่ดัสตัม (full custom)

เกตอาเรย์ (Gate Array)

วงจรรวมนี้ประกอบด้วยแถวลำดับของวงจรเกต ซึ่งอาจจะเป็นวงจรเกตประเภทเดียวกันหรือต่างชนิดกันก็ได้ กับขั้วต่อสายไฟ (Pad) สำหรับต่อกับวงจรภายนอก ผู้ใช้มีหน้าที่ออกแบบการเชื่อม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โยงทางไฟฟ้าระหว่างวงจรเกตแต่ละตัวและขั้วต่อสายไฟเพื่อให้ทำหน้าที่ตามต้องการ แล้วจึงส่งผลการออกแบบนี้ไปยังโรงงานผู้ผลิตวงจรรวมเกตอาเรย์ นั้นไปทำการเจือสารต่อไป โดยทั่วไปไปแล้วปรากฏในการใช้วงจรรวมเกตอาเรย์สร้างวงจรที่ต้องการใช้งานทั่วไปจะมีเพียง 25-30% ของพื้นที่ซิลิคอนที่ใช้สำหรับวงจรเกต ส่วนพื้นที่ที่เหลือจะใช้กับการเชื่อมโยงวงจรเกตเข้าด้วยกันเองและเข้ากับขั้วต่อสายไฟ พื้นที่ซิลิคอนจะใช้ประโยชน์สูงถึง 75% สำหรับวงจรเชิงเลขที่มีลักษณะสม่ำเสมอ เช่น หน่วยความจำเป็นต้น

เซลล์มาตรฐาน (Standard Cell)

ปัญหาการใช้พื้นที่ซิลิคอนอย่างมากมายสำหรับการเชื่อมโยงวงจรของวงจรรวม เกตอาเรย์ทำให้เกิดขีดจำกัดในความซับซ้อนของวงจรที่ใช้ ประกอบกับผู้ใช้จำนวนมากต้องการรวบรวมวงจรรวมมาตรฐาน อาทิ วงจรตระกูล 7400 วงจรตระกูล 4000 จำนวนหลายตัวเข้าเป็นวงจรรวม ASIC เพียงตัวเดียว ทำให้เกิดวงจรรวม ASIC แบบเซลล์มาตรฐานขึ้น วงจรรวม เซลล์มาตรฐานนี้ ผู้ใช้เป็นผู้เลือกกลุ่มวงจรทำหน้าที่ต่างๆ เช่น เกต ฟลิบฟลอป ตัวนับ ตัวเลื่อน หน่วยความจำ หรือกระทั่งไมโครโปรเซสเซอร์จากแฟ้มข้อมูล (Library) ของผู้ผลิต ซึ่งอาจจะเป็นแฟ้มข้อมูลคอมพิวเตอร์ เหมาะสมสำหรับวงจรเชิงเลขที่มีความซับซ้อน การผลิตวงจรรวม เซลล์มาตรฐานจะมีต้นสูงกว่าวงจรรวมเกตอาเรย์และใช้เวลาในการออกแบบและเจือสารยาวกว่าสองถึงสามเท่าตัว วงจรรวมเซลล์มาตรฐานจึงเหมาะสมกับการใช้งานเชิงพาณิชย์ที่มีปริมาณการผลิตนับหมื่นตัวขึ้นไปบทบาทวงจรรวมเซลล์มาตรฐานจะมีเพิ่มมากขึ้นในอนาคต เมื่อตัวทุนการเจือสารลดลง

ฟูลัสตัม (Full Custom)

วงจรฟูลัสตัม นี้ผู้ใช้เป็นผู้ออกแบบเองทั้งหมด ตั้งแต่ระดับวงจรเชิงเลขจนกระทั่งจึงระดับกายภาพ แล้วจึงส่งข้อมูลการออกแบบไปให้ผู้ผลิตเจือสารในรูปแฟ้มข้อมูลมาตรฐาน เช่น GDS II, CIF การใช้ฟูลัสตัม ASIC นี้เท่าที่แพร่หลายอยู่ในปัจจุบัน จะเป็นไปเพื่อการศึกษาและการวิจัยและเพื่อการผลิตจำนวนน้อย ส่วนใหญ่จะเป็นการเจือสารในลักษณะที่ใช้ค่าใช้จ่ายร่วมกันกล่าวคือ รวบรวมการออกแบบกลายวงจรลงบนแผ่นซิลิคอนเดียวกันเพื่อประหยัดค่าเจือสาร

รูปแบบการเจือสารวงจรรวมฟูลัสตัมที่ประกอบด้วยการออกแบบหลายวงจรรวมอยู่บนแผ่นซิลิคอนเดียวกัน มีกระทำอยู่ 3 รูปแบบดังนี้

1. Multi-project wafer (MPW) การเจือสารวงจรรวมฟูลัสตัมแบบนี้ แผ่นเวเฟอร์ (Wafer) จะแบ่งเป็นส่วนๆ ที่เรียกว่าได (Die) และไดแต่ละชุดจะเจือสารวงจรต่างกัน ผู้ผลิตที่ให้บริการแบบนี้ได้แก่ โมซิส (MOSIS) และ ออบิต (ORBIT) ในสหรัฐอเมริกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2. Multi-project chip (MPC) เป็นการเจือสารออกแบบหลายวงจรลงบนไดชุดเดียวกันโดยที่ไดทุกชุดบนเวเฟอร์เดียวกันจะมีลักษณะเหมือนกัน การจัดวางวงจรที่ออกแบบลงบนได เน้นการใช้พื้นที่ซิลิคอนให้เป็นประโยชน์สูงสุดเป็นสำคัญ ตัวอย่างผู้ผลิตที่ให้บริการแบบนี้ได้แก่ อาวา (AWA) ในออสเตรเลีย และ อีเอสทู (ES II) ในสหราชอาณาจักร

3. Multi-project reticle (MPR) เป็นการเจือสารในลักษณะผสมผสานระหว่างฟูลัสตัมกัลเซลล์มาตรฐานกล่าวคือ ได้จะแบ่งออกเป็นส่วนๆ อย่างสม่ำเสมอแต่ละส่วนบรรจุการออกแบบแต่ละวงจร โดยไดชุดหนึ่งจะบรรจุวงจรที่ออกแบบได้ประมาณ 8-วงจรและไดทุกชุดในแผ่นเวเฟอร์จะมีลักษณะเหมือนกัน-ในปัจจุบันมีเพียงอาวาที่ใช้บริการการเจือสารลักษณะนี้ โดยจำกัดเฉพาะประเภทเซลล์มาตรฐานเท่านั้น

3.3 เอฟพีจีเอ (FPGA: Field Programmable Gate Array)

เป็นอุปกรณ์ที่ถูกพัฒนาต่อจากอุปกรณ์แอลซีเอของบริษัทไซรินซ์ (XILINX Inc.) โดยมีประสิทธิภาพการทำงานและมีปริมาณความหนาแน่นของเกตสูง สามารถจะกำหนดฟังก์ชั่นการทำงานได้ความต้องการของผู้ใช้โดยผ่านการโปรแกรมเอฟพีจีเอได้รวบรวมข้อดีทั้งหมดของการทำคัสตัมวีแอลเอสไอ (Custom VLSI) มารวมไว้ทั้งหมดได้แก่ การออกแบบการผลิต, ลดเวลาที่จะส่งตัวผลิตภัณฑ์ออกตลาด ซึ่งเป็นประโยชน์ต่อการผลิตวงจรเป็นอย่างมาก นักออกแบบเพียงกำหนดฟังก์ชั่นการทำงานของวงจร ดังนั้นการออกแบบวงจรโดยใช้เอฟพีจีเอ สามารถออกแบบและทดสอบภายในเวลาเพียง 2-3 วันเท่านั้น ตรงกันข้ามกับการออกแบบโดยใช้เกตอาเรย์ ซึ่งใช้เวลาหลายอาทิตย์การเปลี่ยนแปลงแก้ไขแบบก็เช่นเดียวกัน จากประโยชน์ของเอฟพีจีเอ ดังกล่าวมา ทำให้เกิดการประหยัดค่าใช้จ่ายเป็นอย่างมาก เพราะได้ความเสี่ยงในการที่จะต้องแก้ไขตัววงจร การเลื่อนเวลาการออกผลิตภัณฑ์ ลดค่าเอ็นอาร์อี (NRE : Nonrecurring Engineering Cost) ลงไปด้วย

เทคโนโลยีของ FPGA

เนื่องจากเป็นลักษณะของชิพที่สามารถโปรแกรมได้นั่นก็คือ สามารถกำหนดจุดเชื่อมต่อต่างๆ ภายในได้ เพื่อประกอบเป็นลักษณะของวงจรตามที่เราต้องการได้ ซึ่งเราสามารถแบ่งลักษณะของจุดเชื่อมต่อต่างๆได้ดังนี้

1. Physical Changing

1.1 Fused สามารถโปรแกรมได้เพียงครั้งเดียว หลังจากโปรแกรมจุดเชื่อมต่อขาดจากกัน

1.2 Anty Fuse สามารถโปรแกรมได้เพียงครั้งเดียวหลังจากโปรแกรม จุดเชื่อมต่อจะเชื่อมถึงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50402

2. Memory Base

2.1 EEPROM – Base FPGA

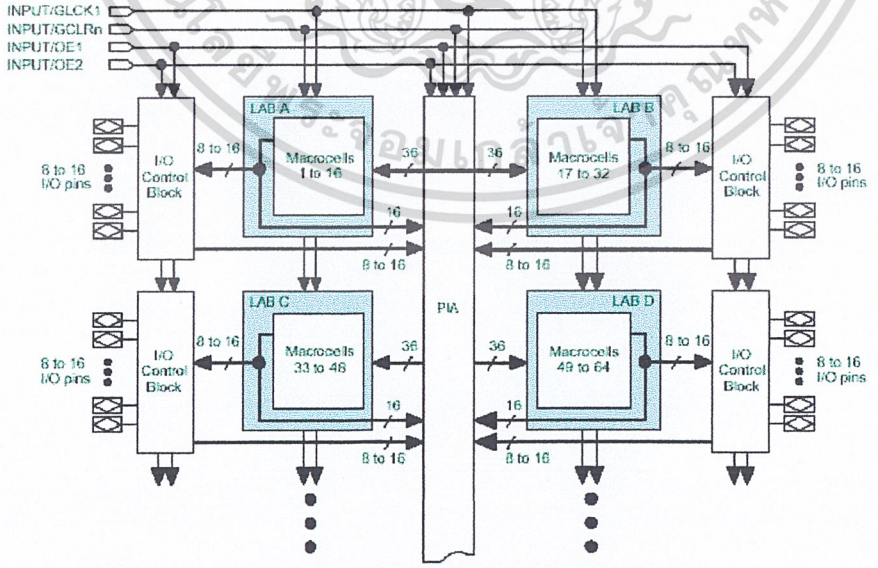
มักเรียก FPGA ประเภทนี้ว่า CPLD จะใช้เทคโนโลยีเหมือนกับ EEPROM ในการโปรแกรม ซึ่งจะทำให้มีความจุของเกตต่ำ โดยทั่วๆไปจะน้อยกว่า 20,000 เกต แต่ข้อดีของ EEPROM-Base FPGA คือสามารถเก็บข้อมูลที่โปรแกรมลงไปได้โดยไม่จำเป็นต้องมีไฟเลี้ยง และในการโปรแกรม จะใช้ทรานซิสเตอร์ 1 ตัวต่อ 1 บิท สามารถโปรแกรมได้ประมาณ 10,000 ครั้ง มักจะมีการจัดสถาปัตยกรรมในรูปแบบอาร์เรย์ ใช้ AND- OR Plane ในการทำลอจิกฟังก์ชัน

2.2 SRAM - Base FPGA

จะใช้เทคโนโลยีเหมือน SRAM ในการโปรแกรม ซึ่งจะสามารถทำให้โปรแกรมซ้ำได้ไม่จำกัดจำนวนครั้ง มีความจุของเกตปานกลางถึงสูงมาก (ประมาณ 10,000 – 1,000,000 เกต) จะใช้ Look-Up Table ในการทำลอจิกฟังก์ชัน (Logic Function) และจะมีการจัดทรัพยากรภายในโครงสร้างแบบอาร์เรย์ ข้อดีของ SRAM - Base FPGA คือจะใช้เวลาในการโปรแกรมน้อย (ในระดับ mS) การโปรแกรมจะทำได้ง่ายเทียบเท่ากับการเขียน SRAM ทั่วไป และไม่จำกัดจำนวนครั้งในกระบวนการผลิตจะทำได้ง่าย และเหมาะสมสำหรับการออกแบบวงจรที่มีความสลับซับซ้อน ข้อเสียก็คือไม่สามารถเก็บโปรแกรมในภาวะที่ไม่มีไฟเลี้ยงได้ มักจะใช้ FPGA ชนิดนี้ควบคู่กับ ROM เพื่อเก็บโปรแกรมและจะโหลดโปรแกรมเข้าในตัวชิพเมื่อเริ่มต้นใช้งาน

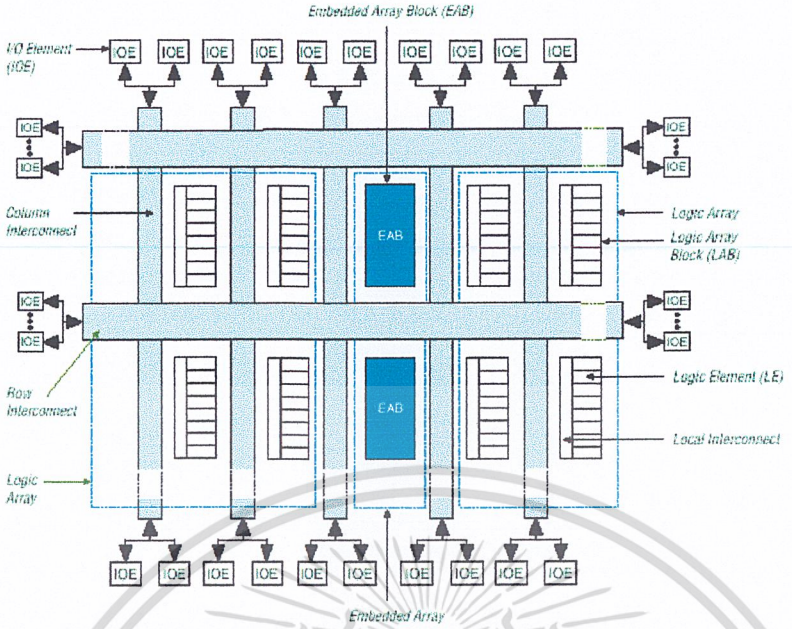
โครงสร้างภายใน

ลักษณะโครงสร้างภายใน เป็นอาร์เรย์ลอจิกบ๊อกซ์ที่สามารถ ทำการโปรแกรม ได้ดังแสดงในรูปที่ 3.5



รูปที่ 3.5 โครงสร้างภายในของ FPGA ตระกูล MAX7000S

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 โครงสร้างภายในของ FPGA ตระกูล FLEK10K

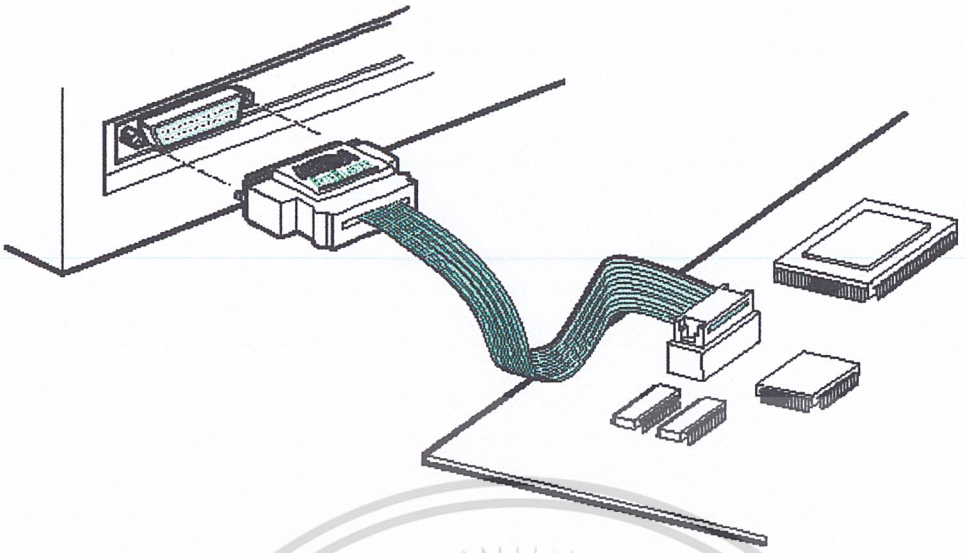
ทำไมการออกแบบถึงทำได้ง่ายและสะดวกรวดเร็ว

1. ในการออกแบบเราไม่จำเป็นต้องรู้ถึงโครงสร้างภายในของตัวชิพ เพียงแต่รู้ขั้นตอนการออกแบบลอจิกก็พอ ไม่เหมือนไมโครโปรเซสเซอร์ที่เราจำเป็นต้องรู้โครงสร้างภายในรวมถึงการศึกษาการเขียนภาษา Assembly ซึ่งแต่ละตัวก็ไม่เหมือนกันด้วย

2. การใช้ภาษาในการอธิบายการทำงานของวงจร ที่เรียกว่า **HDL (Hardware Description Language)** จะช่วยได้มากสำหรับการออกแบบ เนื่องจากเป็นวิธีการที่มีความยืดหยุ่นสูง ทำได้เร็ว และไม่จำเป็นต้องรู้ลักษณะของวงจรที่จะออกแบบว่าต่อกันอย่างไร เพียงแต่กำหนดลักษณะการทำงานให้มัน และตัวซอฟท์แวร์จะทำ Synthesis and Optimize ให้เราเอง นอกจากนี้ภาษาที่ใช้ ยังเป็นมาตราฐานเดียวกัน สามารถใช้ได้กับชิพททุกตัวและทุกบริษัท

3. การโปรแกรมสามารถทำได้เองและใช้เวลาไม่นาน โดยเพียงแค่ส่งข้อมูลผ่านสายดาวโหลดทางพอร์ตของคอมพิวเตอร์ก็สามารถโปรแกรมตัวชิพได้ขณะที่เอยู่ในระบบ โดยไม่จำเป็นต้องถอดมาโปรแกรมข้างนอก ดังรูปที่ 3.7 และที่สำคัญสามารถโปรแกรมได้หลายครั้ง จึงทำให้ง่ายในการแก้ไขและพัฒนาโดยที่ไม่ต้องเสียค่าใช้จ่ายเพิ่มแต่อย่างใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 การโปรแกรมลงในชิพ

การออกแบบโดยใช้ภาษาอธิบายพฤติกรรมของฮาร์ดแวร์ (HDL)

ในการออกแบบวงจรดิจิตอลนั้นสามารถทำได้โดยการวาดวงจร (Schematic) หรือใช้ภาษาอธิบายพฤติกรรม (Hardware Description Language) ของฮาร์ดแวร์ ในกรณีของการออกแบบวงจรด้วย ASIC เราจะต้องเขียนวงจรด้วย Schematic แล้วนำวงจรนั้นไป Simulate หากผลออกมาเป็นที่พอใจก็จะต้อง Layout เป็นชั้นสาร และในการออกแบบ ASIC จำเป็นจะต้องรู้ว่าจะใช้เทคโนโลยีอะไรเช่นที่ NECTEC ใช้อยู่ จะใช้เทคโนโลยีของ ALCATEL 0.5 um. เมื่อได้ชั้น Layout เสร็จสมบูรณ์ จึงจะส่งไป Fabrication เป็นชิพไอซี แล้วจึงจะนำมาใช้งานได้ แต่ในการออกแบบวงจรด้วย FPGA โดยการใช้ Schematic หรือใช้ภาษาอธิบายการทำงานของวงจรจะทำได้สะดวกกว่า เพราะการทำวิธีนี้ ผู้ออกแบบไม่ต้องคำนึงถึงเทคโนโลยีที่จะใช้และที่สำคัญ การออกแบบโดยวิธีนี้สามารถแก้ไขโมเดล (Model) หรือเปลี่ยนแปลงเทคโนโลยีได้สะดวกกว่า เพราะไม่ต้องวาดวงจรใหม่ นั่นคือการออกแบบโดยใช้ภาษาอธิบายฮาร์ดแวร์ จะทำให้โมเดลที่ได้ไม่ขึ้นกับเทคโนโลยี สำหรับภาษาที่ใช้สำหรับอธิบายพฤติกรรมของฮาร์ดแวร์ที่ใช้กันก็มี VHDL, AHDL, Verilogสำหรับในการทดลองนี้จะให้นักศึกษาใช้การวาด **Schematic** ส่วน VHDL นั้นจะมีใน Lab ต่อไป

การสังเคราะห์วงจร (Logic Synthesis)

ในขั้นตอนนี้จะใช้ซอร์พแวร์ในการสังเคราะห์วงจร (Synthesis Tools) ทำการสังเคราะห์พฤติกรรมของวงจรที่ได้จากการออกแบบด้วย Schematic หรือ VHDL ซึ่งต้องทำการตรวจสอบด้วยว่าซอร์พแวร์นั้นสนับสนุนเทคโนโลยี FPGA (FPGA Library) ที่ต้องการหรือไม่ ตัวอย่างเช่น FPGA ของบริษัท

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XILINX และบริษัท ALTERA จะมีซอร์พแวร์หลายตัวที่สามารถใช้ได้ เช่น **Max+Plus II** ในขั้นตอนนี้ ซอร์พแวร์สังเคราะห์วงจรจะทำการแปลงโค้ด VHDL และทำการ Optimize เพื่อให้ได้วงจรตามเทคโนโลยีที่เลือกใช้ ในการสังเคราะห์วงจรนั้นวงจรระดับเกต (Gate Level) จะไม่เหมาะสมกับโครงสร้างทีมีอยู่ในอุปกรณ์ FPGA ดังนั้นในการ Optimize ซอร์พแวร์สังเคราะห์วงจร จะต้องทำการ Optimize ให้ได้เป็นวงจรที่ประกอบด้วยกลุ่มของลอจิกที่เหมาะสมกับอุปกรณ์ FPGA นั้นๆ จึงทำให้ผลที่ได้มีประสิทธิภาพและในขั้นตอนการสังเคราะห์วงจรนี้ ผู้ออกแบบสามารถกำหนดข้อบังคับสำหรับโมเดลแต่ละตัวได้ เช่น ข้อบังคับในเรื่องเวลา (Timing Constraints) หรือข้อบังคับในเรื่องของพื้นที่ (Area) หรือกำหนดชนิดและตำแหน่งของ I/O ซึ่งข้อบังคับเหล่านี้จะถูกนำไปใช้ในขั้นตอน Optimize เพื่อให้วงจรที่ได้เป็นไปตามที่กำหนด ส่วนสำคัญในการ Optimize คือการเทียบ (Mapping) โมเดลให้เข้ากับเทคโนโลยีที่ใช้เพื่อให้ได้วงจรที่เหมาะสมกับโครงสร้างและสถาปัตยกรรมภายในอุปกรณ์ FPGA เมื่อทำการสังเคราะห์วงจรเสร็จแล้ว ซอร์พแวร์ การสังเคราะห์วงจรก็จะมีการรายงานผลว่าโมเดลที่ออกแบบไปนั้นเป็นอย่างไร เช่นมีค่าความหน่วง (Delay) เท่าไหร่ ใช้ทรัพยากรต่างๆใน FPGA อะไรบ้าง เมื่อมาถึงขั้นตอนนี้ ผู้ออกแบบก็จะทราบว่าโมเดลเป็นไปตามข้อบังคับหรือไม่ ถ้าไม่ก็สังเคราะห์ใหม่จนกว่าจะเป็นไปตามที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ภาษา วีเอชดีแอล

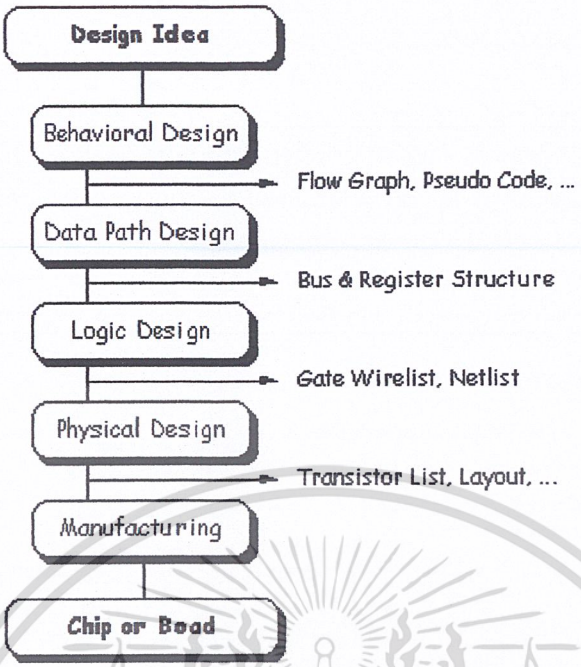
ภาษา วีเอชดีแอล

ความซับซ้อนและขนาดของระบบดิจิตอลในปัจจุบันได้เพิ่มมากขึ้นทุกขณะส่งผลให้มีการนำคอมพิวเตอร์เพื่อช่วยในการออกแบบหรือ CAD มาใช้ในขบวนการออกแบบฮาร์ดแวร์เพิ่มขึ้นเช่นกัน อีกทั้งอุปกรณ์และวิธีการ ออกแบบใหม่ๆ ก็ถูกพัฒนาขึ้นมาเพื่อช่วยอำนวยความสะดวกให้กับนักออกแบบมากขึ้นด้วย สำหรับภาษาบรรยายอุปกรณ์ฮาร์ดแวร์ (HDL : Hardware Description Language) ก็เป็นเครื่องมืออย่างหนึ่งที่ได้รับการพัฒนามาอย่างต่อเนื่อง เพื่อช่วยให้การปรับปรุงขบวนการออกแบบระบบดิจิตอลเป็นไปอย่างมีประสิทธิภาพ

การออกแบบระบบดิจิตอล

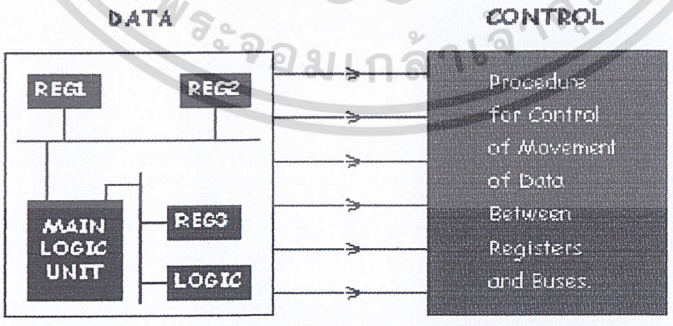
ในการออกแบบระบบดิจิตอล เริ่มตั้งแต่การกำหนดแนวความคิดเบื้องต้นจนกระทั่งได้ออกมาเป็นอุปกรณ์ฮาร์ดแวร์ ที่ใช้งานได้จะต้องผ่านขั้นตอนต่างๆ มากมาย และในแต่ละขั้นตอนผู้ออกแบบจะต้องตรวจสอบผลลัพธ์ในแต่ละขั้น ก่อนเข้าสู่กระบวนการออกแบบในขั้นต่อไป รูปที่ 3.1 แสดงขั้นตอนปกติที่ใช้ในการออกแบบระบบดิจิตอลทั่วไป ขั้นแรกผู้ออกแบบจะกำหนดแนวความคิดในการออกแบบแล้วทำการพัฒนาให้สามารถนำมาใช้ได้อย่างสมบรูณ์ ซึ่งภายในขั้นตอนนี้ผู้ออกแบบจำเป็นต้องสร้างรูปแบบระบบในเชิงพฤติกรรมขึ้นมาตรวจสอบซึ่งอาจจะเป็นผังงานแสดงแบบหรือรหัสคำสั่งเทียม (Pseudo code) ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 แสดงขั้นตอนการออกแบบระบบดิจิตอล

ขั้นตอนต่อไปเป็นการออกแบบระบบเส้นทางของข้อมูล ผู้ออกแบบจะกำหนดส่วนประกอบของรีจิสเตอร์และวงจรลอจิก ที่จำเป็นทั้งหมดเพื่อนำมาประกอบเป็นระบบที่สมบูรณ์ โดยแต่ละองค์ประกอบสามารถเชื่อมต่อกันด้วยบัสหนึ่งหรือสอง ทิศทาง (Unidirectional or Bidirectional Bus) ส่วนกระบวนการในการควบคุมการเคลื่อนย้ายข้อมูลระหว่าง รีจิสเตอร์และวงจรลอจิกจะขึ้นอยู่กับพฤติกรรมของระบบที่กำหนดไว้ดังรูปที่ 4.2



รูปที่ 4.2 การออกแบบระบบเส้นทางของข้อมูล

ขั้นตอนถัดมาเป็นการออกแบบวงจรลอจิก ซึ่งจะเกี่ยวข้องกับการนำเกทดิจิตอลพื้นฐานและฟลิปฟลอป (flip-flop) มาประกอบเป็นอุปกรณ์ย่อยต่างๆ เช่น รีจิสเตอร์เก็บข้อมูล บัสวงจรลอจิก และส่วนควบคุมฮาร์ดแวร์ ซึ่งผลลัพธ์ที่ได้ในขั้นตอนนี้จะเป็นเครือข่ายของการโยงใยระหว่างเกทและ

ฟลิปฟลอปนั่นเองการออกแบบในขั้นตอนถัดไป เป็นการเปลี่ยนเครือข่ายการโยงใยที่ได้จากขั้นตอนที่แล้วให้เป็นลำดับของทรานซิสเตอร์ (Transistor List) และ Layout ซึ่งขั้นตอนนี้จะเกี่ยวข้องโดยตรงกับการจัดวางทรานซิสเตอร์หรือไลบรารีเซลล์เพื่อ แทนเกทและฟลิปฟลอปต่างๆและในขั้นตอนสุดท้ายจะเป็นการส่งระบบที่ออกแบบไว้ไปทำการเจือสารที่โรงงานเพื่อผลิตออกมาเป็น วงจรรวมในที่สุด

4.1 ประวัติความเป็นมาของภาษา VHDL

VHDL ย่อมาจากคำว่า VHSIC Hardware Description Language (VHSIC : Very High Speed Integrated Circuit) เป็นภาษาโปรแกรมระดับสูง (High Level Language) ที่ใช้สำหรับการออกแบบฮาร์ดแวร์ในระบบดิจิตอล ตัวของภาษาสามารถบรรยายพฤติกรรมการทำงานในรูปของลำดับชั้น (Hierarchy) และ สามารถเขียนได้หลายรูปแบบ ด้วยเหตุผลนี้จึงทำให้ภาษา VHDL เป็นเครื่องมือที่ใช้ออกแบบตั้งแต่ขั้นตอนบนสุด คือ แนวความคิดที่จะแก้ปัญหา ลงไปทีละขั้นจนถึงขั้นตอนของการสร้างวงจรจริง และตัวภาษาก็เปิดโอกาสให้วิศวกร ได้พัฒนาและจำลองการทำงานของรูปแบบฟังก์ชันการทำงานของวงจรอย่างสังเขป โดยยังไม่ต้องคำนึงถึงรายละเอียดเกี่ยวกับโครงสร้างวงจรจริง นอกจากนั้น VHDL ยังเป็นภาษาที่สนับสนุนลักษณะต่างๆ ของระบบดิจิตอลที่มี ความซับซ้อนได้ทั้งหมด ดังนั้น VHDL จึงเป็นภาษาที่น่าสนใจในการศึกษาและนำไปใช้งานเป็นอย่างยิ่ง วิวัฒนาการของภาษา VHDL เริ่มต้นประมาณปี ค.ศ. 1981 เมื่อกระทรวงกลาโหมสหรัฐอเมริกา หรือ DoD (Department of Defense) ได้พยายามปรับปรุงอุปกรณ์อิเล็กทรอนิกส์และคอมพิวเตอร์ที่ใช้ในกิจการทางทหาร ให้มีความทันสมัยมากขึ้น ประกอบกับเทคโนโลยีทางด้านไมโครอิเล็กทรอนิกส์มีการพัฒนาไปอย่างรวดเร็วดังจะเห็นได้จากการนำวงจรดิจิตอลหลายๆ วงจรมาทำการผลิตอยู่บนแผ่นซิลิกอนที่มีพื้นที่เพียง 1 - 2 ตารางเซนติเมตรเท่านั้น ซึ่ง เป็นผลให้ประสิทธิภาพในการทำงานของวงจรสูงขึ้นตลอดจนความน่าเชื่อถือ ในการทำงานและความคงทนต่อสภาพ แวดล้อมสูง แต่เนื่องจากในขณะนั้นขั้นตอนของการออกแบบ การผลิต และการตรวจสอบวงจรต้นแบบ เป็นขบวนการที่ ต้องใช้วิศวกร และเวลาในดำเนินการมาก ฉะนั้น ทาง DoD จึงจัดตั้งโครงการขึ้นมาเพื่อศึกษาวิธีการที่ช่วยในการพัฒนา วงจรอิเล็กทรอนิกส์ โดยเฉพาะอย่างยิ่งวงจรระบบดิจิตอล ให้สามารถนำไปผลิตได้เร็วขึ้น ซึ่งโครงการดังกล่าวมีชื่อว่า "Very High Speed Integrated Circuits" หรือ VHSIC โดยในระยะแรกนั้นโครงการนี้ถือเป็นความลับทาง ด้านความมั่งคงของประเทศ และอยู่ภายใต้ความควบคุมดูแลของ United States International Traffic and Arms Regulations (ITAR) สำหรับมาตรฐานของภาษาที่ใช้บรรยาย พฤติกรรมวงจรหรือฮาร์ดแวร์ของระบบสำหรับโครงการ VHSIC ที่ DoDได้ให้ไว้สามารถสรุปได้ดังนี้

- ต้องเป็นภาษาที่นำไปเขียนรูปแบบระบบดิจิตอล และมีคุณสมบัติที่สามารถเข้าใจได้ทั้งมนุษย์และเครื่องคอมพิวเตอร์ โดยไม่ ต้องมีการแปลหรือเปลี่ยนแปลงอีก

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถนำไปใช้เป็นเอกสารประกอบโครงการได้
- ต้องเป็นภาษาที่เขียนขึ้นสำหรับใช้จำลองการทำงานของวงจร

ฉะนั้นภาษาดังกล่าวนี้จึงจัดเป็นภาษาโปรแกรมระดับสูง เช่นเดียวกับภาษาปาสคาล หรือภาษาซี ซึ่งในทางวิศวกรรม ภาษาที่ใช้ในการออกแบบฮาร์ดแวร์นี้เรียกว่า "Hardware Description Language" หรือ HDL

ในตอนเริ่มแรกนั้น DoD ได้มอบหมายให้บริษัทไอบีเอ็ม เท็กซัสอินสตูเมนท์ และอินเตอร์เมทริกซ์ เป็นผู้ศึกษาและพัฒนาโครงการ ซึ่งการดำเนินงานเป็นไปอย่างต่อเนื่อง จนกระทั่งในปี ค.ศ.1985 ทาง ITAR ได้ยกเลิกข้อจำกัดในการถ่ายทอด เทคโนโลยีทางทหารออกจากโครงการนี้ ดังนั้นภาษา VHDL จึงเริ่มเป็นที่รู้จักกันโดยทั่วไป และประมาณปี ค.ศ. 1987 IEEEได้ทำการกำหนดมาตรฐานของภาษานี้เป็น 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 VHDLเท่านั้น ซึ่งทำให้ DoD สามารถนำโครงการเหล่านี้ไปจำลองกับเครื่องคอมพิวเตอร์ได้ หลายๆระบบ

ข้อกำหนด

DoD ได้ตั้งข้อกำหนดสำหรับภาษา VHDL ในเดือนมกราคมปี ค.ศ.1983 ไว้ดังนี้

1. ลักษณะทั่วไป

DoD ได้กำหนดให้ VHDL เป็นภาษาสำหรับการออกแบบและบรรยายของฮาร์ดแวร์ ซึ่งหมายถึงความสามารถ ในการอธิบายและออกแบบในระดับสูง การจำลอง (Simulation) การสังเคราะห์ (Synthesis) และการทดสอบ (Testing) นอกจากนั้น VHDLยังถูกกำหนดไว้สำหรับการบรรยายฮาร์ดแวร์ตั้งแต่ระดับบน ซึ่งก็คือระบบจนถึง ระดับเกทอีกด้วย เนื่องจากในการทำงานของระบบดิจิตอลนั้น ทุกๆ องค์ประกอบภายในระบบไม่ว่าเล็กหรือใหญ่จะทำงานไปพร้อมๆ กัน ซึ่งในเรื่องของความพร้อมเพรียงในการทำงานนี้ก็ถือเป็นข้อกำหนดที่สำคัญอย่างหนึ่งของ VHDL ด้วยเช่นกัน (สำหรับในภาษาที่ใช้ในการบรรยายฮาร์ดแวร์นั้นความพร้อมเพรียงจะหมายถึงทุกๆ คำสั่ง องค์ประกอบ เกทหรือวงจรลอจิกจะถูกนำมาปฏิบัติทั้งหมด ดังนั้นในที่สุดแล้วก็จะดูเหมือนว่าได้มีการปฏิบัติไป พร้อมๆ กัน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. สนับสนุนการออกแบบแบบลำดับขั้น

การออกแบบแบบลำดับขั้น เป็นลักษณะที่สำคัญอย่างหนึ่ง สำหรับการออกแบบระบบที่มีหลายๆ ระดับ โดยในการออกแบบจะประกอบด้วย ส่วนการบรรยายการเชื่อมต่อ และส่วนการบรรยายหน้าที่การทำงาน ซึ่งหน้าที่การทำงานของระบบสามารถกำหนดได้ด้วยตัวเอง หรือ อาจถูกกำหนดโดยโครงสร้างที่ประกอบด้วยองค์ประกอบย่อยๆ ลงไปได้เช่นกัน แต่ที่ระดับล่างสุด องค์ประกอบต้องถูกบรรยายหน้าที่การทำงานด้วยตัวมันเอง และ ไม่สามารถกำหนดการทำงาน โดยลักษณะแบบโครงสร้างได้

3. ไลบรารี

VHDL ได้สนับสนุนการมีไลบรารี เพื่อระบบการจัดการที่ดี ผู้ออกแบบสามารถกำหนดลักษณะและการทำงานของอุปกรณ์พื้นฐานไว้ในระบบไลบรารี หรือจะใช้ไลบรารีที่ระบบได้จัดเตรียมไว้แล้วก็ได้ โมเดลและการบรรยายที่ถูกต้องควรจัดเก็บไว้ในไลบรารี หลังจากที่ได้ผ่านการคอมไพล์เรียบร้อยแล้ว เพื่อให้ผู้ออกแบบคนอื่นๆ สามารถนำไป ใช้ได้ด้วย

4. ลำดับคำสั่ง

แม้ว่าการปฏิบัติคำสั่งหรือกระบวนการ โดยพร้อมเพรียงกันจะเป็นคุณสมบัติที่สำคัญของ VHDL ก็ตาม ตัวภาษาเองก็ยังมีการจัดเตรียมลักษณะการควบคุมแบบลำดับคำสั่งไว้ให้ด้วย เมื่อผู้ออกแบบได้กำหนดหน้าที่และองค์ประกอบที่ทำงานพร้อมกันของระบบไว้เรียบร้อยแล้ว ผู้ออกแบบยังสามารถบรรยายหน้าที่การทำงาน ซึ่งเป็นรายละเอียดภายในของแต่ละองค์ประกอบได้ในลักษณะเดียวกับการเขียนโปรแกรมที่ประกอบด้วย โครงสร้างแบบ case, if - then - else และ loop ทั่วๆ ไปได้ การบรรยายแบบลำดับคำสั่งทำให้การออกแบบหน้าที่การทำงานของอุปกรณ์กระทำได้สะดวก และง่ายขึ้นอย่างไรก็ตามโครงสร้างทั้งหมดของ VHDL ก็ยังคงเป็นการทำงานแบบพร้อมเพรียงกันเช่นเดิม

5. การกำหนดคุณสมบัติ

นอกจากการกำหนดอินพุทและเอาท์พุทแล้ว เงื่อนไขอื่นๆ ก็มีผลต่อการปฏิบัติหน้าที่ของอุปกรณ์ฮาร์ดแวร์ด้วยเช่นกัน โดยสิ่งนี้รวมถึงสภาพแวดล้อม และ ลักษณะทางกายภาพของอุปกรณ์นั้นๆ ด้วยซึ่งภาษาสำหรับการออกแบบที่ดีควรให้ผู้ออกแบบกำหนดคุณสมบัติของอุปกรณ์ที่ใช้ได้ด้วย เช่น สามารถกำหนดขนาด ลักษณะทางกายภาพเวลา โหลด และเงื่อนไขทางสภาพแวดล้อมอื่นๆซึ่งความสามารถในการกำหนดคุณสมบัตินี้ก็เป็นส่วนหนึ่งที่มีอยู่ในภาษา VHDL ด้วยเช่นกัน

6. ชนิดของข้อมูล

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. โปรแกรมย่อย

ความสามารถในการใช้ฟังก์ชันและโพรซีเจอร์ (Procedure) ก็เป็นข้อกำหนดอีกอย่างหนึ่งใน VHDL ซึ่งผู้ออกแบบสามารถนำโปรแกรมย่อยมาใช้ในการเปลี่ยนแปลงชนิดของข้อมูล การกำหนดหน่วยของลอจิก การกำหนดตัวกระทำต่างๆ หรือหน้าที่อื่นๆ ตามที่ต้องการได้เช่นเดียวกับการเขียนโปรแกรมทั่วไป

8. การควบคุมเวลา

VHDL อนุญาตให้ผู้ออกแบบสามารถกำหนดเวลาในการส่งผ่านข้อมูลหรือสัญญาณได้ตามต้องการ การตรวจสอบ การออกแบบเกทหรือการหน่วงเวลาก็สามารถกระทำได้โดยการกำหนดช่วงเวลาที่แน่นอน หรือกำหนดให้มีการรอคอย เหตุการณ์ (Event) นอกจากนี้ก็ยังสามารถกำหนดรูปแบบของสัญญาณนาฬิกาได้อีกด้วย

9. การกำหนดแบบโครงสร้าง

การกำหนดโครงสร้างขององค์ประกอบต่างๆ สามารถกระทำได้ในทุกระดับของการออกแบบ โดยการกำหนดโครง สร้างขององค์ประกอบร่วมที่เกิดจากองค์ประกอบย่อย ซึ่งแตกต่างกันหรือ เหมือนกันก็เป็นข้อกำหนดอย่างหนึ่งของ VHDL เช่นกัน

4.2 องค์ประกอบพื้นฐานของ VHDL

ในการเขียนรูปแบบบรรยายระบบดิจิตอลในมุมมองของการออกแบบลักษณะบนลงล่าง จะต้องทำความเข้าใจในเรื่องของโครงสร้างและส่วนประกอบต่างๆ ของรูปแบบภาษาวีเอชดีแอลเสียก่อน ซึ่งส่วนประกอบที่สำคัญและเป็นพื้นฐานของการเขียนมี 4 หน่วยคือ

- หน่วยการออกแบบเอนทิตี (Entity Design Unit)
- หน่วยการออกแบบสถาปัตยกรรม (Arcjotectire Design Unit)
- หน่วยการออกแบบแพ็กเก็ต (Paclage Design Unit)
- หน่วยการออกแบบโครงแบบ (Configuration Design Unit)

หน่วยการออกแบบเอนทิตี

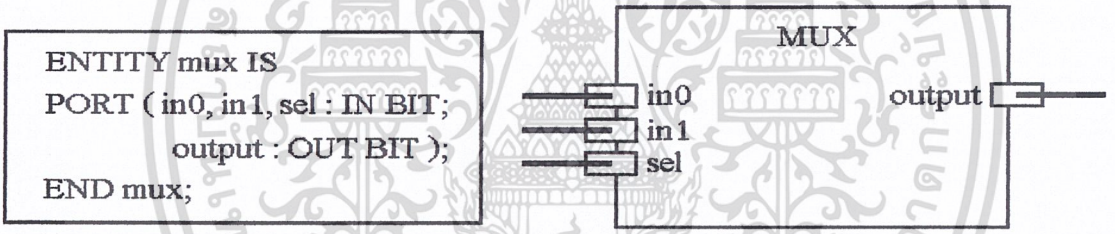
หน่วยการออกแบบนี้เป็นส่วนที่ใช้สำหรับติดต่อระหว่างโลกภายนอกกับรูปแบบที่เขียนขึ้น ที่เรียกว่า หน่วยการออกแบบเอนทิตี ในส่วนนี้ใช้กำหนดจุดเชื่อมต่อ ของรูปแบบ กำหนดทิศทางการไหลของสัญญาณ และประเภทของค่าที่สามารถกำหนดให้กับสัญญาณตามจุดต่างๆ ของข้อมูลที่ไหล่ผ่านจุดต่อเหล่านั้น รูปที่ 4.3 แสดงให้เห็นโครงสร้างอย่างง่าย ๆ ของ หน่วยการออกแบบเอนทิตี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

รูปที่ 4.3 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี

ส่วนนี้จะขึ้นต้นด้วยคำ ENTITY และ IS ระหว่างคำทั้งสองเป็นส่วนสำหรับชื่อของรูปแบบที่ต้องการจะเขียน (component_name) สำหรับการตั้งชื่อนั้นต้องเป็นไปตามกฎเกณฑ์ของภาษาหลังจากนั้นจะตามด้วยส่วนที่ใช้กำหนดช่องทางเข้าและออกของข้อมูล (input-output) รวมทั้งพารามิเตอร์อื่นๆ ส่วนนี้เรียกว่าส่วนหัว(entity header) และที่สำคัญคือ หน่วยการออกแบบเอนทิตีจะต้องปิดท้ายด้วยคำว่า END และเครื่องหมายอัฒภาคเสมอ (;)

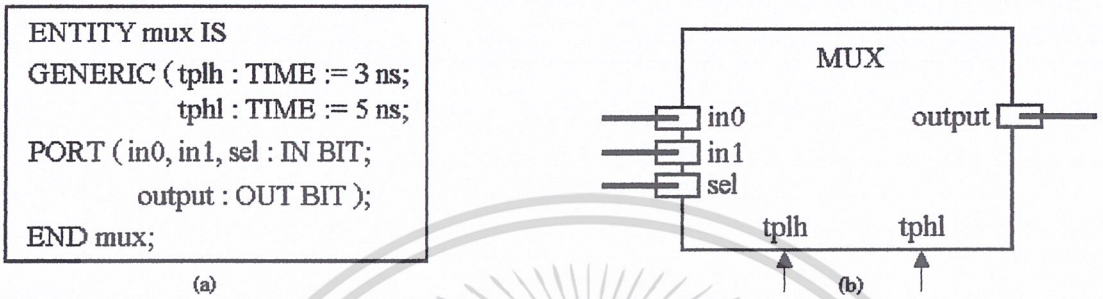


รูปที่ 4.4 แสดงรูปแบบของมัลติเพลกซ์
(a) หน่วยการออกแบบเอนทิตีในรูปของวีเอชดีแอล
(b) มุมมองของตัวเชื่อมประสาน (Interfacing)

ในรูปที่ 4.4 เป็นหน่วยการออกแบบเอนทิตี ที่บรรยายอุปกรณ์ที่มีชื่อว่ามัลติเพลกซ์ หรือ MUX ในส่วนหัวของเอนทิตี มีการกำหนดจัดต่อ 4 จุดภายใต้ชุดคำสั่ง PORT โดยที่ 3 จุดแรกเป็นจุดให้ข้อมูลไหลผ่านเข้า ได้แก่ in0.in1.sel ซึ่งกำหนดด้วยทิศทางการติดต่อกับภายนอกเป็นการไหลเข้าของข้อมูล (IN) ที่แสดงด้วยรูปสี่เหลี่ยมโปร่งในรูปที่ 4.4 ส่วนจุดเอาต์พุตเป็นจุดให้ข้อมูลไหลออก ซึ่งกำหนดด้วยทิศทางการติดต่อกับภายนอกเป็นการไหลออก (OUT) ที่แสดงด้วยรูปสี่เหลี่ยมทึบในรูปที่ 4.4 ส่วนประกอบประเภทของข้อมูลที่จะไหล เข้าและออก นั้นเป็นประเภท BIT ที่สามารถมีค่าได้เพียงสองค่าคือ “0” และ “1” เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนั้นผู้ออกแบบยังสามารถกำหนดค่าพารามิเตอร์ทางฟิสิกส์ที่เป็นข้อมูลเพิ่มเติมอื่นๆ ลงในส่วนหัวของเอนทิตีได้อีก เช่น ข้อมูลเกี่ยวกับความเร็วในการทำงานของอุปกรณ์ อันได้แก่ ค่าเวลาหน่วงแพร่กระจาย (Propagation delay time) พารามิเตอร์เหล่านี้ เรียกว่า เจนเนริก (Generic) ที่กำหนดด้วยคำสั่ง GENERIC จากตัวอย่างในรูปที่ 4.5



รูปที่ 4.5 รูปแบบมัลติเพลกซ์ที่ประกอบด้วยข้อมูลค่าเวลาหน่วงแพร่กระจาย
(a) หน่วยการออกแบบเอนทิตีในรูปของวีเอชดีแอล
(b) มุมมองของตัวเชื่อมประสาน

ในบางกรณีสามารถที่จะใช้ภาษาวีเอชดีแอล สร้างรูปแบบที่ปราศจากช่องทางไหลเข้าและออกของข้อมูลได้ ซึ่งส่วนใหญ่จะพบในการสร้างรูปแบบ สำหรับตรวจสอบการทำงานของอีกรูปแบบหนึ่ง คือ วีเอชดีแอลสำหรับการทดสอบเปรียบเทียบ (Test bench)

```
ENTITY test_bench
IS
END test bench;
```

รูปที่ 4.6 หน่วยการออกแบบเอนทิตีที่ไม่มีการกำหนดช่องทางที่ต่อกับภายนอก

หน่วยการออกแบบสถาปัตยกรรม

คือส่วนที่ใช้เขียนบรรยายพฤติกรรมของรูปแบบ ในมุมมองของการจำลองการทำงานพฤติกรรมต่างๆ ที่บรรยายในส่วนนี้ขึ้นอยู่กับข้อมูลที่ผ่านเข้าและออก ตรงช่องทางตลอดจนพารามิเตอร์ต่างๆ ที่กำหนดใน หน่วยการออกแบบเอนทิตี รูปที่ 4.7 แสดงให้เห็นถึงโครงสร้างอย่างง่ายๆ ของหน่วยการออกแบบสถาปัตยกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ARCHITECTURE *identifier* OF
component_name IS
[declaration
BEGIN
specification of the functionality of the component in terms of its input lines and as influenced by physical and other parameters
END *[identifier];*
parameters
END *[identifier];*

รูปที่ 4.7 หน่วยการออกแบบเอนติตี้ที่ไม่มีการกำหนดช่องทางที่ต่อกับภายนอก

ส่วนของหน่วยการออกแบบสถาปัตยกรรมเริ่มต้นด้วยคำ ARCHITECTURE และตามด้วยชื่อ (identifier) สิ่งที่ต้องกำหนดลงไปได้แก่ สิ่งที่แสดงให้เห็นว่า ARCHITECTURE นั้นใช้ บรรยายหน่วยการออกแบบเอนทิตีใดๆ (OT<entity design unit> IS) ส่วนที่อยู่ระหว่าง ARCHITECTURE และ BEGIN เป็นพื้นที่ส่วนประกาศหน่วยของสถาปัตยกรรมกำหนด (architecture declarative area) ที่เป็นเพียงส่วนเผื่อเลือก (option) ในบริเวณนี้สามารถใช้เขียนประกาศกำหนดค่าต่างๆ ที่จะนำไปใช้ภายในสถาปัตยกรรมนั้นได้ อาทิเช่นประเภท (type) ต่างๆ ตัวอย่างเช่น bit, bit_vector), สัญญาณ (signal), ตัวคงที่ (constant) โปรแกรมย่อย ได้แก่ function และ procedure) และอุปกรณ์ (component) ส่วนที่ใช้บรรยายความสัมพันธ์ระหว่างข้อมูลที่ไหลเข้า และไหลออกของรูปแบบ (สัญญาณที่กำหนดในชุดคำสั่ง PORT) นั้นจะถูกบรรยายในบริเวณเนื้อที่ระหว่างคำว่า BEGIN กับ END ของหน่วยการออกแบบสถาปัตยกรรม และนอกจากนั้นชุดคำสั่งทุกคำสั่งที่อยู่ภายในบริเวณนี้จะเป็นชุดคำสั่งแบบแข่งขนาน (concurrent statement) เท่านั้น หน่วยการออกแบบสถาปัตยกรรมจะต้องปิดท้ายด้วยคำสั่ง END และชื่อของสถาปัตยกรรมนั้น ๆ เป็นส่วนเผื่อเลือกโดยทั่วไปการเขียนรูปแบบระบบดิจิตอลด้วยภาษาวีเอชดีแอลสามารถเขียนได้ในลักษณะต่างๆ ดังนี้

- ประเภทการไหลของข้อมูล (Dataflow description)
- ประเภทพฤติกรรม (Behavioral description)
- ประเภทโครงสร้าง (Structure description)
- ประเภทผสม (Mixed model description)

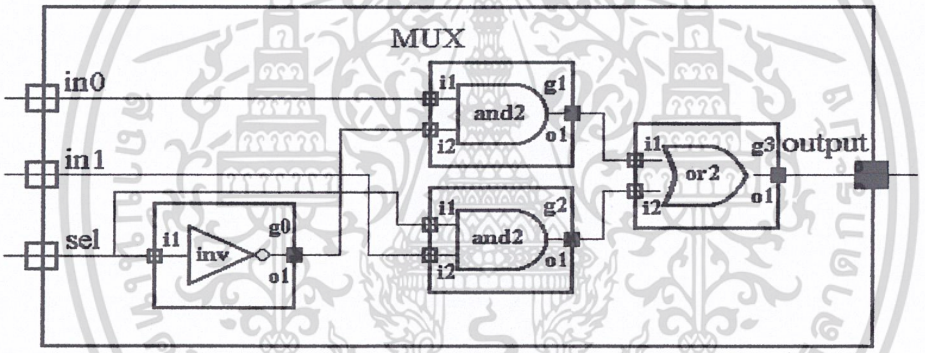
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ARCHITECTURE data_flow OF mux IS
BEGIN
Output <= ((NOT sel) AND in0) OR (sel AND
in1);
END data_flow;
```

รูปที่ 4.8 แสดงหน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ตามฟังก์ชันบูลีน

Output = (sel.in0) + (sel.in1)

จากรูปที่ 4.8 ส่วนที่บรรยายความสัมพันธ์ระหว่างข้อมูลที่ไหลเข้า (*in0, in1*) กับข้อมูลที่ไหลออก (output) ประกอบด้วยชุดคำสั่งแบบแข่งขันขนานเพียงชุดเดียว ซึ่งเขียนเป็นประเภทการไหลของข้อมูลของมัลติเพลกซ์ หรือ ระดับการถ่ายโอนข้อมูลระหว่างรีจิสเตอร์ (RTL:Register Transfer Level)



รูปที่ 4.9 แสดงโครงสร้างภายในสถาปัตยกรรมของมัลติเพลกซ์

จากรูปที่ 4.9 เป็นหน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทโครงสร้าง โดยใช้ อินเวอร์เตอร์ (inv ที่ตำแหน่ง g0), แอนด์เกต 2 อินพุตจำนวน 2 ตัว (and2 ที่ตำแหน่ง g1 และ g2) และ ออร์เกต 2 อินพุต (or2 ที่ตำแหน่ง g3) มาสร้างตามฟังก์ชันบูลีนของรูปที่ 4.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ARCHITECTURE struc OF mux IS
COMPONENT inv
PORT (i1 : IN  BIT ; o1 : OUT  BIT);
COMPONENT and2
PORT (i1, i2 : IN BIT ; o3 : OUT  BIT);
COMPONENT or2
PORT (i1, i2: IN BIT; o1: OUT BIT);
END COMPONENT;
SIGNAL int0, int1, int2: BIT:
BEGIN
g0 : inv        PORT MAP (i1 => sel, o1 =>
int0);
g1 : and2     PORT MAP (i1 => in0, i2 => int0,
o1  => int1);
g3 : or2        PORT MAP (i1 => in1, i2 =>
int2, o1 => ouput );
```

รูปที่ 4.10 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทโครงสร้าง

```
ARCHITECTURE behav  OF mux  IS
BEGIN
PROCESS (in0, in1, sel)
BEGIN
      IF (sel = '0") THEN  output <=
in0:
      ELSE  output <= in1;
      END  IF;
END   PROCESS
END  behav;
```

รูปที่ 4.11 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทพฤติกรรม

ไม่ว่าเขียนบรรยายส่วนของสถาปัตยกรรมของมัลติเพลกซ์ในลักษณะของ ประเภทพฤติกรรม ประเภทการไหลของข้อมูล ประเภทโครงสร้างหรือประเภทผสมที่นำเอาแต่ละประเภทมาเขียนไว้ในส่วนของสถาปัตยกรรม ก็ตามต่างก็มีพฤติกรรมเดียวกันและจะให้ผลลัพธ์จากการจำลองการทำงานที่เหมือนกัน ซึ่งนี้ก็เป็นข้อดีของภาษาวีเอชดีแอล

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบพื้นฐานที่ใช้ในการบรรยายถึงองค์ประกอบของ VHDL จะประกอบไปด้วยส่วนกำหนดการเชื่อมต่อ (Interface) และส่วนกำหนดลักษณะเชิงสถาปัตยกรรม (Architecture) ดังแสดงในรูปที่ 4.12 โดยในการบรรยายการเชื่อมต่อจะขึ้น ต้นด้วยคำว่า ENTITY แล้วตามด้วยชื่อขององค์ประกอบจากนั้นตามด้วยคำว่า IS และถัดมาจะเป็นการบรรยายถึงพอร์ต การติดต่อ อินพุท - เอาท์พุท ขององค์ประกอบ ส่วนลักษณะภายนอกอื่น ๆ เช่น เวลา อุณหภูมิก็สามารถรวมเข้าไปในส่วนนี้ ได้เช่นกัน ในส่วนของการกำหนดลักษณะเชิงสถาปัตยกรรมจะขึ้นต้นด้วยคำว่า ARCHITECTURE ซึ่งเป็นส่วนที่ใช้บรรยายหน้าที่การทำงานขององค์ประกอบ โดยหน้าที่การทำงานนี้จะขึ้นอยู่กับสัญญาณอินพุท เอาท์พุท และพารามิเตอร์ อื่นๆ ที่ได้กำหนดไว้ในส่วนของการเชื่อมต่อดังรูปที่ 4.12 และสำหรับการบรรยายหน้าที่ขององค์ประกอบจะเริ่มต้นหลังจากคำว่า BEGIN เป็นต้นไป

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

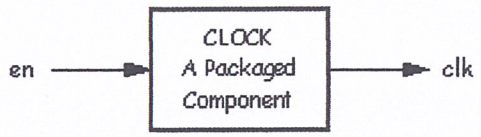
ARCHITECTURE identifier OF component_name IS
    [declartion]
BEGIN
    specification of the functionality of the component
    in terms of its input lines and as influenced
    by physical and other parameters
END [identifier];
```

รูปที่ 4.12 การกำหนดการเชื่อมต่อและสถาปัตยกรรม

การกำหนดการเชื่อมต่อ

การกำหนดการเชื่อมต่อเป็นระดับบนสุดของการออกแบบโดยในระดับนี้ต้องกำหนดพอร์ตสำหรับการติดต่อกับองค์ประกอบภายนอกอื่นๆ ดังตัวอย่างในรูปที่ 4.13 ซึ่งเป็นบล็อคไดอะแกรมและการบรรยายการเชื่อมต่อขององค์ประกอบ สำหรับตัวจ่ายสัญญาณนาฬิกาในบรรทัดแรกของการบรรยายการเชื่อมต่อเป็นการกำหนดชื่อขององค์ประกอบ ซึ่งกำหนดเป็น clock_ component ตามด้วยคำว่า PORT และชื่อของพอร์ตอยู่ภายในวงเล็บ ส่วน IN และ OUT เป็นการกำหนดโหนดของสัญญาณให้เป็นอินพุทหรือเอาท์พุท และ BIT เป็นการแสดงชนิดของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




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

รูปที่ 4.13 บล็อคไดอะแกรมและการบรรยายการเชื่อมต่อของ clock_ component

การกำหนดรูปแบบการบรรยาย

หน้าที่การทำงานขององค์ประกอบจะถูกบรรยายภายในส่วนนี้ ซึ่งในการบรรยายสามารถกำหนดค่าของสัญญาณเอาท์พุทในเทอมของอินพุทหรือในรูปขององค์ประกอบอื่นๆ หรือทั้งสองอย่างรวมกันก็ได้ดังตัวอย่างการบรรยายของ Clock_component ในรูปที่ 4.14 ซึ่งเป็นการบรรยายในเชิงพฤติกรรมโดยมี en เป็นอินพุทและ ck เป็นเอาท์พุท PROCESS เป็นคำที่ใช้ในการเริ่มต้นสำหรับการบรรยายในเชิงพฤติกรรม และภายใน โปรเซสกำหนดให้ periodic เป็นตัวแปรที่มีค่าเริ่มต้นเป็น "0" ถ้าสัญญาณ en มีค่าเป็น "1"' จะทำให้ตัวแปร periodic ถูกคอมพลีเมนท์ (complement) และส่งค่าให้กับ ck ซึ่งเป็นสัญญาณเอาท์พุท และสำหรับคำสั่ง WAIT จะเป็นการกำหนดให้สัญญาณมีคาบเวลาเท่ากับ 1 ไมโครวินาที

```
ARCHITECTURE behavioral OF clock_component IS

BEGIN
   PEOCESS
      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 4.14 การบรรยายเชิงพฤติกรรมของ clock _ component

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยการออกแบบแพ็กเก็ต

ข้อมูลต่างๆ ตลอดจนโปรแกรมย่อยที่เป็นประโยชน์ต่อการเขียนรูปแบบ การบรรยายระบบดิจิตอล สามารถเก็บไว้ใน ส่วนของแพ็กเก็ต ซึ่งหน่วยการออกแบบต่างๆ เช่น หน่วยการออกแบบ Entity หน่วยการออกแบบสถาปัตยกรรมหรือ หน่วยการออกแบบแพ็กเกจอื่นๆ สามารถเรียกข้อมูลเหล่านี้ไปใช้ได้ นอกจากนั้นสิ่งที่นิยมทำกันมากคือการนำรูปแบบ มาตรฐานต่างๆ เช่น อุปกรณ์มาตรฐาน (เช่น ไอซีตระกูล 74XX เป็นต้น)มาเก็บไว้ในรูปของแพ็กเกจ ที่ทุกคนสามารถ เข้าถึงได้ ตามปกติแล้วแพ็กเกจจะแบ่งออกเป็น 2 ส่วนคือ การประกาศแพ็กเกจ (Package declaration)และ ส่วนของบอดี้แพ็กเกจ (Package body) เนื่องจาก แพ็กเกจถูกสร้างขึ้นเป็นส่วนแยกต่างหากออกจากรูปแบบที่กำลังเขียนอยู่ ฉะนั้นการที่นำแพ็กเกจไปใช้นั้นจะต้องมีการเชื่อมโยงหรืออ้างอิงเสียก่อน ซึ่งในภาษา VHDL สามารถ กระทำได้ด้วยชุดคำสั่ง USE

- PACKAGE DECLARATION

ส่วนที่มีความสำคัญที่สุดของแพ็กเกจ (ถ้ามองในแง่ของการนำไปใช้จากภายนอก) ได้แก่ ส่วนการประกาศแพ็กเกจ เนื่องจากเป็นส่วนที่ใช้กำหนดชื่อของสิ่งที่ประกาศอยู่ภายในแพ็กเกจสำหรับนำไปใช้ภายนอกตัวของแพ็กเกจเอง ถ้ามี การประกาศสิ่งใดๆ ในส่วนของส่วนบอดี้แพ็กเกจแต่ไม่ถูกประกาศในส่วนการประกาศแพ็กเกจจะทำให้ค่าและพฤติกรรมไม่สามารถนำไปใช้งานในส่วนนอกได้ ซึ่งเปรียบเทียบได้กับสิ่งที่ประกาศไว้ในส่วนของการประกาศ Entity คือ จุดเชื่อมต่อหรือ พอร์ต ที่มีหน้าที่ติดต่อกับโลกภายนอก ฉะนั้นโดยทั่วไปแล้วแพ็กเกจสามารถสร้างขึ้นได้โดยไม่จำเป็นต้องมีส่วนบอดี้ และยังสามารถนำไปใช้งานจากรูปแบบภายนอกได้เช่น ใช้สำหรับประกาศ ชนิด (Type) หรือสัญญาณ เช่นเดียวกับ ส่วนบอดี้แพ็กเกจที่ไม่จำเป็นต้องมี ส่วนของการประกาศแพ็กเกจแต่แพ็กเกจนั้นจะไม่สามารถนำไปใช้จาก รูปแบบอื่นได้

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

รูปที่ 4.15 โครงสร้างทั่วไปของส่วนการประกาศแพ็กเกจ

- PACKAGE BODY

โครงสร้างซึ่งประกอบด้วยลำดับคำสั่งที่ใช้บรรยายฟังก์ชันการทำงานของโปรแกรมย่อยทั้งหลาย ซึ่งชื่อของโปรแกรมย่อยนั้นๆ ได้ถูกประกาศไปแล้ว ในส่วนของการประกาศแพ็กเกจ จะถูกเก็บไว้ในส่วนของบอดี้แพ็กเกจ ทั้งนี้รวมถึง การกำหนดค่าคงที่ต่างๆ อันได้แก่ค่าคงที่ที่ถูกประกาศ

ชื่อไว้ก่อน ในส่วนของการประกาศแพ็กเกจ และถูกกำหนดค่าใน ส่วนของบอดี้แพ็กเกจ ฉะนั้นในส่วนของบอดี้แพ็กเกจจึงไม่จำเป็นต้องมี ถ้าในส่วนของการประกาศแพ็กเกจไม่มีการ ประกาศชื่อที่เป็นโปรแกรมย่อย หรือค่าคงที่ การเขียนบอดี้แพ็กเกจนั้นจะเป็นไปตามกฎเกณฑ์ดังแสดงในรูปที่ 4.16

```
PACKAGE BODY package_name IS
    declarative part
END package_name;
```

รูปที่ 4.16 โครงสร้างของบอดี้แพ็กเกจ

หน่วยการออกแบบ Configuration

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

```
COMFIGURTRION identifier OF entiry_name IS
    Configuretion_declarative_part
END ;
```

รูปที่ 4.17 โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ

โปรแกรมย่อย

การใช้ฟังก์ชันและโพรซีเจอร์ใน VHDL เปรียบได้กับการใช้โปรแกรมย่อยในการเขียนโปรแกรมภาษาชั้นสูงทั่วๆ ไปค่าที่ถูกส่งกลับ หรือถูกเปลี่ยนแปลงโดยโปรแกรมย่อยอาจจะมีหรือไม่มีผลต่อฮาร์ดแวร์โดยตรงก็ได้ เช่นถ้าใช้ฟังก์ชัน แทนการกระทำในสมการบูลีนก็จะมีผลต่อวงจรลอจิกจริงๆ ในขณะที่ถ้าใช้โปรแกรมย่อยในการเปลี่ยนชนิดของข้อมูล หรือในการคำนวณค่าการหน่วงเวลาแล้วก็จะไม่มีผลต่อโครงสร้างของฮาร์ดแวร์ รูปที่ 2.18 แสดงการใช้โพรซีเจอร์ เพื่อเปลี่ยนข้อมูลชนิด 8 บิตเป็นค่าจำนวนเต็ม และรูปที่ 3.19 แสดงการใช้ฟังก์ชันโดยกำหนดให้ X เป็นตัวแปรชนิด บิตแทนการกระทำในสมการบูลีน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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;
                END IF;
        END LOOP;
        oi := result;
END byte_to_integer
```

รูปที่ 4.18 การใช้โพรซีเจอร์

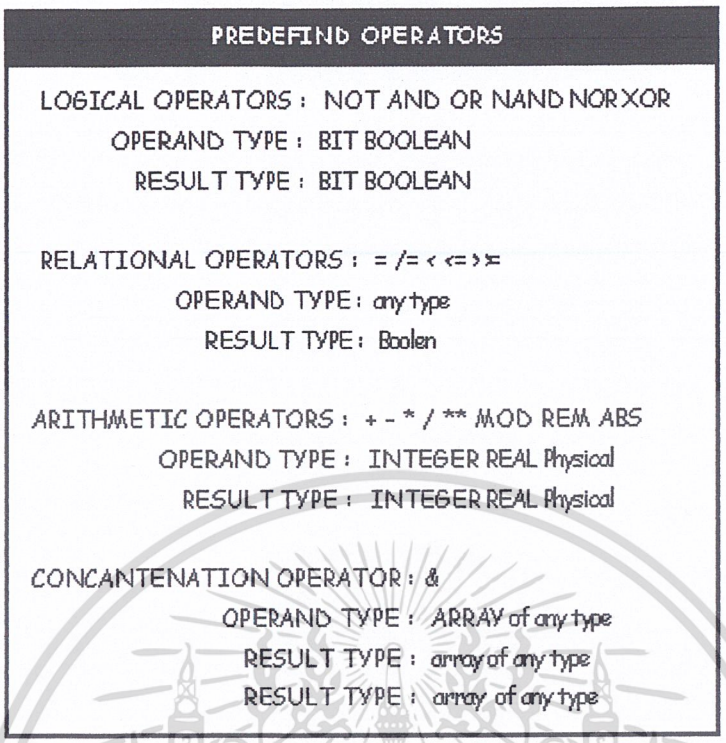
```
FUNCTION f (a, b, c:BIT) RETURN BIT  IS
        VARIABLE x: BIT;
BEGIN
        x := ((NOT a) AND (NOT b) AND c);
        RETURN x;
END f;
```

รูปที่ 4.19 การใช้ฟังก์ชัน

โอเปอร์เรเตอร์

การบรรยายเชิงพฤติกรรมในภาษาVHDLมีตัวดำเนินการหรือ โอเปอร์เรเตอร์ทางลอจิกและคณิตศาสตร์เช่นเดียว กับภาษาซอฟต์แวร์ทั่วไปดังรูปที่ 4.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 ตัวดำเนินการใน VHDL

เวลาและความพร้อมเพรียง

ในวงจรอิเล็กทรอนิกส์อุปกรณ์ทุกๆ ตัวจะอยู่ในสภาพเตรียมพร้อมเสมอ (Always Active) และจะมีเรื่องของเวลาเข้ามาเกี่ยวข้องในทุกๆเหตุการณ์ที่เกิดขึ้นเสมอ VHDL เป็นภาษาที่ได้รับการออกแบบมาเพื่อให้สามารถบรรยายรูปแบบและการพ้องกันของเวลาสำหรับการทำงานของอุปกรณ์ได้อย่างถูกต้อง การบรรยายการทำงานที่อยู่ภายในส่วน ของการบรรยายสถาปัตยกรรม จะมีการทำงานที่พร้อมเพรียงกันเสมอ หรือแม้แต่โปรเซสซึ่งมีการทำงานภายในเป็น แบบลำดับคำสั่งก็ตาม ซึ่งหากมีหลายๆ โปรเซสอยู่ภายในโครงสร้างเดียวกัน ทุกๆ โปรเซสก็จะทำงานไปพร้อมๆ กัน ด้วย

สัญญาณและตัวแปร

สัญญาณมีลักษณะเป็นเสมือนตัวกลางฮาร์ดแวร์ที่ใช้ในการส่งผ่านข้อมูล และ มีเรื่องของเวลาเข้ามาเกี่ยวข้องด้วยการ กำหนดค่าให้กับสัญญาณจะใช้สัญลักษณ์ <= ในการส่งค่าและสามารถใช้คำสั่ง AFTER เพื่อกำหนดช่วงเวลาในการ ส่งผ่านค่าของสัญญาณ เช่น w <= a AFTER 12 NS หมายถึงการกำหนดค่าสัญญาณ a ให้กับ w หลังจากเวลา ผ่านไป 12 นาโนวินาที ในทางตรงข้ามตัวแปรมีลักษณะเป็นเสมือนตัวกลางที่ใช้ในการส่งผ่านข้อมูลและไม่มีเรื่องของ เวลาเข้ามาเกี่ยวข้องด้วย ซึ่งตัวแปรจะถูกใช้ในส่วนที่มีการทำงานเป็นแบบลำดับคำสั่งเช่นใน ฟังก์ชัน โพรซีเจอร์ และ โปรเซส สำหรับการกำหนดค่าให้กับตัวแปรจะใช้สัญลักษณ์: =

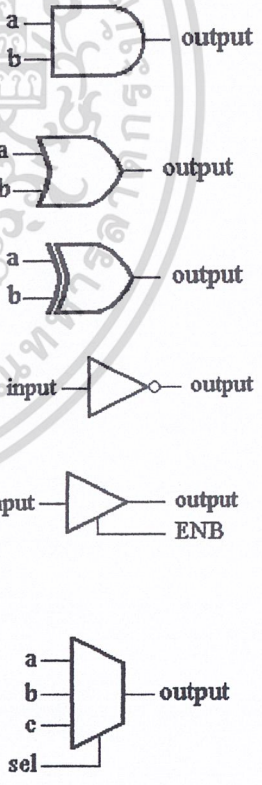
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ภาษาวีเอชดีแอลเพื่อการสังเคราะห์

ภาษาวีเอชดีแอล เป็นภาษาที่เขียนเพื่อจำลองการทำงานของวงจร ซึ่งในบางรูปแบบการเขียนไม่สามารถที่จะนำไปสังเคราะห์ได้ทั้งหมด ดังนั้นถ้าต้องการเขียนเพื่อนำไปสังเคราะห์ ควรหลีกเลี่ยงรูปแบบต่างๆ ที่ไม่สามารถนำไปสังเคราะห์ได้ ในที่นี้ขึ้นอยู่กับความสามารถของโปรแกรมที่ใช้สังเคราะห์แต่ละโปรแกรม ดังนั้นในหัวข้อนี้จะแสดงตัวอย่างของการเขียนโมเดลในรูปแบบต่างๆที่สามารถนำไปสังเคราะห์ ซึ่งยึดหลักการเขียนตาม ViewSynthesis User's Guide ของโปรแกรมซึ่งเป็นโปรแกรม Viewlogic ที่ใช้ในสังเคราะห์วงจรทั้งหมดในการออกแบบไมโครคอนโทรลเลอร์โดยแบ่งออกเป็น 2 กลุ่ม คือ เกตและฟลิปฟลอปประเภทต่างๆ

4.3.1 ตัวอย่างรูปแบบการเขียนเกตพื้นฐาน

```
SIGNAL a, b, c, d, input : vlbit_1d(3 DOWNTO 0);
SIGNAL se1 : vlbit_1d(1 DOWNTO 0);
SIGNAL enb : vlbit;
.....
        ouput <= a AND b;
.....
        ouput <= a OR b;
.....
        ouput <= a XOR b;
.....
        ouput <= a NOT b;
....
ouput <= a input WHEN enb='1' ELSE "ZZ";
....
WITH se1 SELECT
        output <= a WHEN "00" ELSE
                  b WHEN "01" ELSE
                  c;
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 ตัวอย่างรูปแบบการเขียนฟลิปฟลอปพื้นฐาน

```
SIGNAL input, output : vlbit_ld (3 DOWNTO 0);
SIGNAL clock, enable, reset : vlbit ;
    ....
    PROCESS BEGIN
        WAIT UNTIL PRISING (clock);
            Output <= input;
    END PROCESS;
    ....
    PROCESS BEGIN
    WAIT UNTIL PRISING (clock) OR (reset='1');
      IF (reset='1') THEN output <='00';
      ELSIF (enble = '1' ) THEN ouput <= input ;
      END IF ;
    END PROCESS;
```

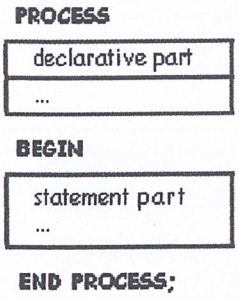
4.4 การบรรยายเชิงพฤติกรรม

การบรรยายลักษณะการทำงานของอุปกรณ์ฮาร์ดแวร์ในเชิงพฤติกรรม เป็นการบรรยายลักษณะการเปลี่ยนแปลงของ ข้อมูลในรูปแบบของอัลกอริธึม สำหรับการคำนวณผลลัพธ์ที่เกิดขึ้น ซึ่งสืบเนื่องมาจากการเปลี่ยนแปลงสภาวะของข้อมูล ที่เข้ามาโดยไม่คำนึงถึงลักษณะโครงสร้างหรือความสัมพันธ์ของอุปกรณ์ที่อยู่ภายในว่าจะเป็นอย่างไรในหัวข้อนี้จะ แสดงถึงการบรรยายเชิงพฤติกรรม แทนการใช้โมดูลฮาร์ดแวร์รวมถึงข้อกำหนดต่างๆ ที่ควรรู้

โปรเซส

โปรเซสเป็นรูปแบบพื้นฐานอย่างหนึ่งที่ใช้ในการกำหนดให้กับสัญญาณ โปรเซสจะอยู่ในสถานะที่เตรียมพร้อมอยู่เสมอ และจะปฏิบัติคำสั่งพร้อมๆกันกับโปรเซสอื่นๆ ที่อยู่ในสถาปัตตกรรมบรรยายเดียวกัน โดยโปรเซสจะปฏิบัติงานตามคำสั่งทันทีที่มีเหตุการณ์เกิดขึ้นกับญญาณที่อยู่ทางด้านขวามือของสัญลักษณ์กำหนดค่าให้กับสัญญาณ (<=) การบรรยาย โปรเซสจะเริ่มต้นด้วยคำสั่ง PROCESS และจบด้วยคำสั่ง END PROCESS ในรูปที่ 3.21 เป็นการแสดงส่วน ประกอบของการบรรยายแบบโปรเซส ซึ่งประกอบด้วยส่วนของการประกาศตัวแปรที่ต้องใช้และส่วนของการปฏิบัติ คำสั่งเพื่อให้ได้ผลลัพธ์ที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21รูปแบบของการบรรยายแบบโปรเซส

การกำหนดตัวดำเนินการภายในโปรเซส

ตัวดำเนินการภายในโปรเซสมี 3 ชนิดคือ ตัวแปร (Variable) ไฟล์ (File) และตัวคงที่ (Constant) ซึ่งตัวดำเนิน การทั้งสามชนิดนี้หากมีการประกาศไว้ในโปรเซสใดก็จะใช้ได้เฉพาะภายในโปรเซสนั้นเท่านั้นสำหรับการติดต่อกับภาย นอกหรือระหว่างโปรเซสสามารถทำได้โดยใช้สัญญาณ (Signal) หรือตัวคงที่ที่ได้ประกาศไว้ในส่วนของ ARCHITECTURE ในรูปที่ 4.13 แสดงตัวอย่างการประกาศตัวกระทำภายในโปรเซส ซึ่งจะอยู่ระหว่างคำสั่ง PROCESS และ BEGIN และค่าเริ่มต้นที่ถูกกำหนดให้กับตัวดำเนินการภายในโปรเซสจะถูกนำมาใช้ในตอนเริ่มต้น ของการปฏิบัติเพียงครั้งเดียวเท่านั้น ต่างกับค่าเริ่มต้นที่อยู่ภายในโปรแกรมย่อยจะถูกนำมาใช้ทุกครั้งที่มีการเรียกใช้โปรแกรมย่อยนั้น ๆ

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

รูปที่ 4.22 ตัวอย่างการประกาศตัวดำเนินการภายในโปรเซสการกำหนดการกระทำภายในโปรเซส

การกระทำใดๆ ภายในโปรเซสจะเป็นการปฏิบัติแบบลำดับ (Sequential) เสมอ ซึ่งภายในโปรเซสสามารถใช้ประโยค เงื่อนไขหรือการทำซ้ำได้เช่น IF-THEN - ELSE, CASE - WHEN, FOR LOOP และ WHILE LOOP ดังตัวอย่างในรูปที่ 4.23 และ 4.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ARCHITECTURE demo OF partial_process IS
 ....
BEGIN
  PROCESS
  ....
   BEGIN
          ....
          x <= '1';
          IF x = '1' THEN
            perform action_1
          ELSE
            perform action_2
          END IF;
          ...
  END PROCESS;
END demo;
```

รูปที่ 4.23 เงื่อนไขการกระทำในโปรเซส

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

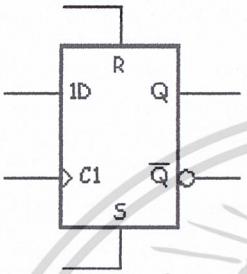
รูปที่ 4.24 แสดงการกระทำในโปรเซส

การกระตุ้นและยับยั้งการกระทำของโปรเซส

การกระทำภายในโปรเซสจะอยู่ในสภาพเตรียมพร้อม และมีการปฏิบัติงานอยู่ตลอดเวลาที่มีการเปลี่ยนแปลงของเหตุการณ์ เกิดขึ้น อย่างไรก็ดีเราสามารถกระตุ้นหรือยับยั้งการกระทำภายในโปรเซสได้โดยการกำหนดรายการของสัญญาณที่ต้อง การให้โปรเซสปฏิบัติงานเมื่อมีเหตุการณ์เกิดขึ้นกับสัญญาณที่กำหนดไว้เท่านั้น ส่วนเหตุการณ์ใดๆ ที่เกิดขึ้นกับสัญญาณ ที่ไม่ได้กำหนดไว้ในรายการก็จะไม่ส่งผล ให้มีการกระทำภายในโปรเซส ซึ่งรายการของสัญญาณนี้เรียกว่า Sensitivity List และจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดไว้ภายในวงเล็บหลังคำสั่ง PROCESS รูปที่ 4.25 (a) แสดงตัวอย่างโมเดล และรูปที่ 4.25 (b) เป็นตัวอย่างการบรรยายการเชื่อมต่อของ D-Flip Flop ส่วนรูปที่ 4.26 แสดงถึงการบรรยายเชิงพฤติกรรมของ D-Flip Flop โดยในรูปที่ 4.26 (a) เป็นการใช้ตัวกระทำภายนอกโปรเซส และรูปที่ 4.26 (b) เป็นการใช้ตัวกระทำภายในโปรเซส โดยมีรายการของสัญญาณ (rst, set, clk) เป็นตัวกระตุ้นการปฏิบัติงานภายในโปรเซส



(a)

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

(b)

รูปที่ 4.25 แสดง D-Flip Flop

(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 rq_delay;
          ELSIF clk = '1' AND clk ' EVENT THEN
            state <= d AFTER cq_delay;
        END IF;
   END PROCESS dff;
   q <= state;
   qb <= NOT state;
END behavioral;
```

(a)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ARCHITECTURE average_dalay_ 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;
      END IF;
         q <= state AFTER (sq_delay + rq_delay + cq_delay)/3;
        qb <= NOT state AFTER(sq_delay + rq_delay + cq_delay)/3;
  END PROCESS dff;
END behavioral;
```

(b)

รูปที่ 4.26 การบรรยายเชิงพฤติกรรมของ D-FlipFlop

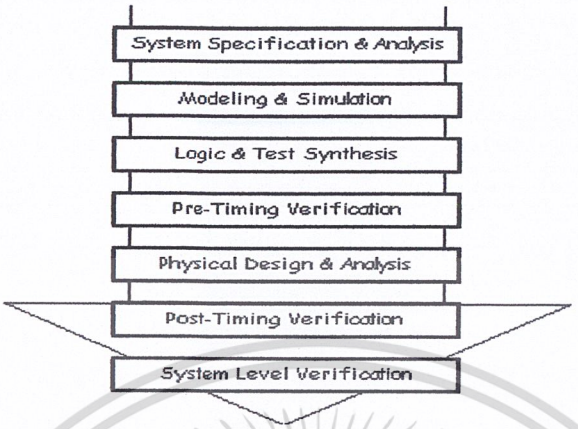
(a) การใช้ตัวกระทำภายนอกโปรเซส

(b) การใช้ตัวกระทำภายในโปรเซส

4.5 การออกแบบจากบนลงล่าง

ในการพัฒนาวงจรรวมดิจิตอลขนาดใหญ่ที่มีความซับซ้อน วิศวกรหรือผู้ออกแบบมักจะมองการออกแบบให้อยู่ในรูปของ บล็อกไดอะแกรมก่อนที่ทำวิเคราะห์ให้ลึกถึงรายละเอียดต่อไป ซึ่งภาษา VHDL นั้นอนุญาตให้อธิบาย และวิเคราะห์การทำงานของแต่ละบล็อก รวมถึงการปรับปรุงการทำงานจากผลที่วิเคราะห์เพื่อให้ได้การทำงานตามต้องการ นอกจากนี้ยังสามารถเพิ่มเติมในรายละเอียดในแต่ละขั้นตอนได้ ซึ่งหลักการนี้สอดคล้องกับหลักการออกแบบจากบนลงล่าง (Top - Down Design) นั่นเอง ถ้าทดลองเปรียบเทียบกับการออกแบบจากล่างขึ้นบน (Bottom - Up Design) จะเห็นได้ว่าการออกแบบจากล่างขึ้นบนจะใช้เวลาการออกแบบมากกว่า 90% เนื่องจากเป็นการวาดวงจรด้วยอุปกรณ์ต่างๆ (Schematic capture) ที่ประกอบกันเข้าเป็นวงจรที่ต้องการออกแบบก่อนแล้วจึงทำการจำลองการทำงาน และตรวจ สอบความถูกต้อง ดังนั้นการใช้ภาษา VHDL กับหลักการออกแบบจากบนลงล่างจึงเป็นทางออกให้กับวิศวกรให้สามารถออกแบบและพัฒนาวงจรที่มีความซับซ้อนได้มากขึ้น ทั้งยังช่วยลดเวลาและค่าใช้จ่ายในการออกแบบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.27 ขั้นตอนการออกแบบจากบนลงล่าง

จากรูปที่ 4.27 แสดงถึงขั้นตอนของการออกแบบจากบนลงล่าง ทั้งนี้ในทางปฏิบัติอาจมีข้อแตกต่างไปจากนี้บ้าง เล็กน้อยเนื่องจากขั้นตอนของการผลิต (Implementation) สามารถกระทำได้หลายเทคโนโลยี สำหรับรายละเอียดของขั้นตอน การออกแบบจากบนลงล่างในแต่ละขั้นตอนมีดังนี้

1. สร้างข้อกำหนดของความต้องการ และวิเคราะห์ระบบ เพื่อหาแนวความคิดและหลักการ (Idea and Concept) ใน การแก้ปัญหา

2.เขียนรูปแบบของระบบที่ต้องการออกแบบ โดยใช้ภาษา VHDL หรือ ภาษา HDL อื่นๆ สำหรับบรรยายพฤติกรรมการ ทำงาน พร้อมทั้งจำลองการทำงาน เพื่อเปรียบเทียบและตรวจสอบความถูกต้องกับข้อกำหนด

3.หลังจากที่ได้หลักการขั้นต้นพร้อมแนวความคิดที่ผ่านการตรวจสอบแล้ว หลักการนี้จะถูกเพิ่มเติมในรายละเอียดลงมา เป็นลำดับขั้นที่สอง จนกระทั่งอยู่ในระดับที่จะนำไปผลิตวงจริง หรือสังเคราะห์ในขั้นตอนนี้เองเทคโนโลยีที่จะมารองรับ วงจรออกแบบจะถูกกำหนดขึ้น และระบบช่วยการออกแบบจะสังเคราะห์วงจรที่ได้จากรูปแบบที่เขียนขึ้นให้อยู่ในรูปของวงจรที่ประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์ หรือวงจรในระดับเกท และการเชื่อมต่อระหว่างกันของอุปกรณ์เหล่านั้นหรือ ไม่ก็อยู่ในรูปของ Netlist ที่สามารถนำไปผลิตในอุปกรณ์อื่นได้

4. หลังจากการสังเคราะห์วงจรให้อยู่ในระดับเกทหรือ Netlist แล้ว ข้อมูลนี้จะถูกใช้สำหรับจำลองการทำงานในเรื่อง ความถูกต้องของฟังก์ชัน พร้อมกับนำข้อมูลที่เกี่ยวข้องกับเวลาเข้ามาประกอบการพิจารณาด้วย ซึ่งตามปกติแล้วอุปกรณ์ ทางอิเล็กทรอนิกส์ทุกชิ้นจะมีเวลาหน่วงของการแพร่กระจาย (Propagation Delay Time) เสมอ ถึงแม้ว่าจะเป็น เวลาที่น้อยมากในระดับนาโนวินาทีก็ตาม แต่ถ้าภายในวงจรหนึ่งประกอบด้วยเกทของฟังก์ชันต่างๆ จำนวน 10,000 เกท ขึ้นไป เวลาดัง

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล่าวนี้จะสะสมกันมากขึ้น จนอาจทำให้การทำงานของวงจรรวมทั้งหมดผิดพลาดไป หรือไม่สามารถทำ งานในย่านความถี่สัญญาณนาฬิกาที่สูงได้

5. ผลิตเป็นวงจรจริง (Technology and device mapping) โดยนำข้อมูลที่ได้จากการสังเคราะห์มาผลิต ซึ่งอาจ จะอยู่ในรูปของแผงวงจรไฟฟ้า ที่ประกอบด้วยอุปกรณ์หลายๆ ชิ้น หรืออยู่ในรูปของวงจรรวม ASIC

6. ทำการตรวจสอบการทำงานและตัวแปรทางด้านเวลาทั้งหมด เพื่อความถูกต้องของวงจรเป็นครั้งสุดท้ายก่อนนำไปรวมเข้ากับอุปกรณ์อื่นๆ ให้เป็นระบบดิจิตอล เนื่องจากในขั้นตอนนี้ วงจรที่ออกแบบ จะประกอบด้วยจุดต่อทางอินพุทและเอาท์พุท ซึ่งเป็นจุดต่อสำหรับการรับและส่งสัญญาณกับภายนอก

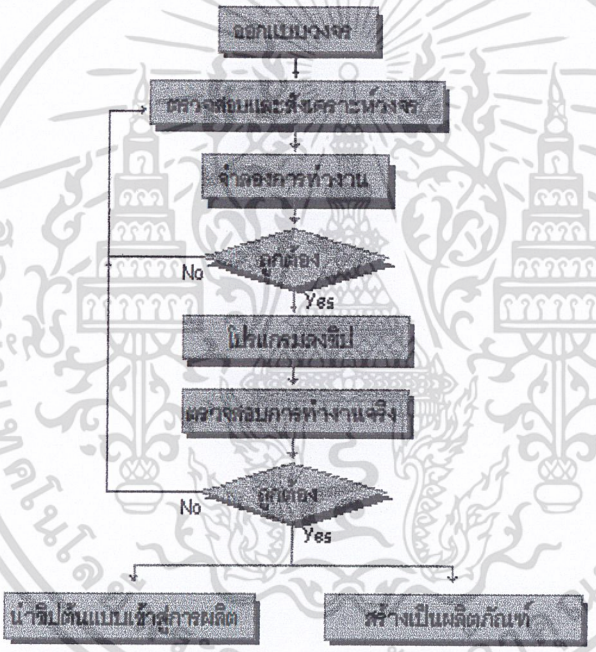
7. นำวงจรที่ออกแบบไว้ประกอบเข้ากับอุปกรณ์อื่นๆ ให้เป็นระบบที่สมบูรณ์ แล้วทำการทดสอบการทำงานทั้งระบบร่วมกับอุปกรณ์อื่นๆ อีกครั้งเพื่อควบคุมคุณภาพของผลิตภัณฑ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การใช้งานโปรแกรม MAX+PLUSE II เบื้องต้น

MAX+PLUSE II เป็นโปรแกรมซึ่งถูกสร้างและพัฒนา ขึ้นมาโดยบริษัท ALTERA เพื่อใช้สำหรับออกแบบ สังเคราะห์วงจร ทางดิจิตอล และ โปรแกรมวงจรลอจิกที่สร้างขึ้นลงในชิพ FPGA ในการออกแบบวงจรด้วย โปรแกรม MAX+PLUSE II สามารถทำได้หลายลักษณะ เช่น

- ใช้การกำหนดรูปแบบ Waveform ทางอินพุทและเอาท์พุทของวงจรที่ต้องการ
- วาดวงจรโดยการนำเกทหรือสัญลักษณ์ต่างๆ มาเชื่อมต่อกัน
- ใช้ภาษาอธิบายพฤติกรรมของวงจรที่ต้องการ เช่น VHDL , AHDL และ Verilog เป็นต้น



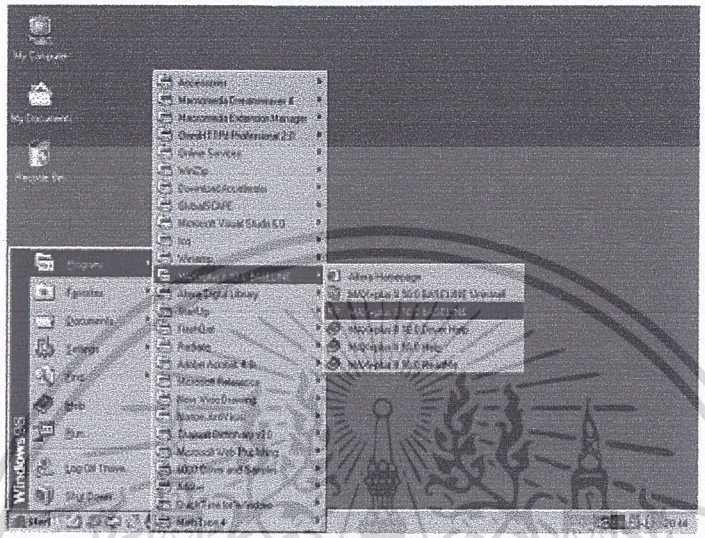
รูปที่ 5.1

ในการออกแบบวงจรด้วยโปรแกรม MAX+PLUSE II สามารถทำได้ดังรูปที่ 1 ซึ่งในส่วนนี้จะอธิบายตั้งแต่ขั้นตอนการออกแบบวงจร, การตรวจสอบความถูกต้องในการต่อวงจร, การคอมไพล์ - สังเคราะห์วงจร, การจำลองการทำงานของวงจรที่ออกแบบมา, การโปรแกรมวงจรลงในชิพและตรวจสอบการทำงานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

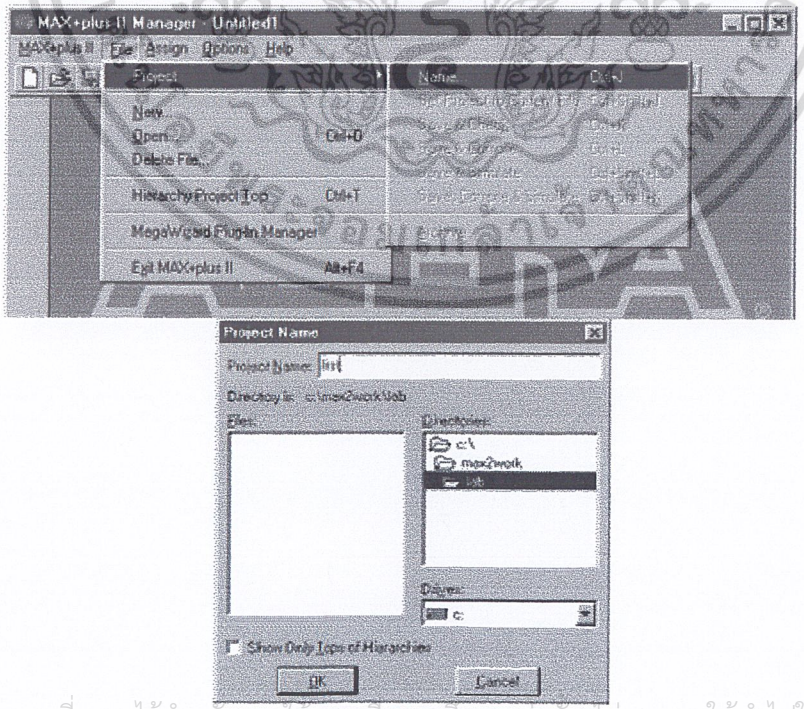
การออกแบบวงจร (Design Entry)

1. เรียกโปรแกรม MAX+PLUSE II จาก Start Menu



รูปที่ 5.2

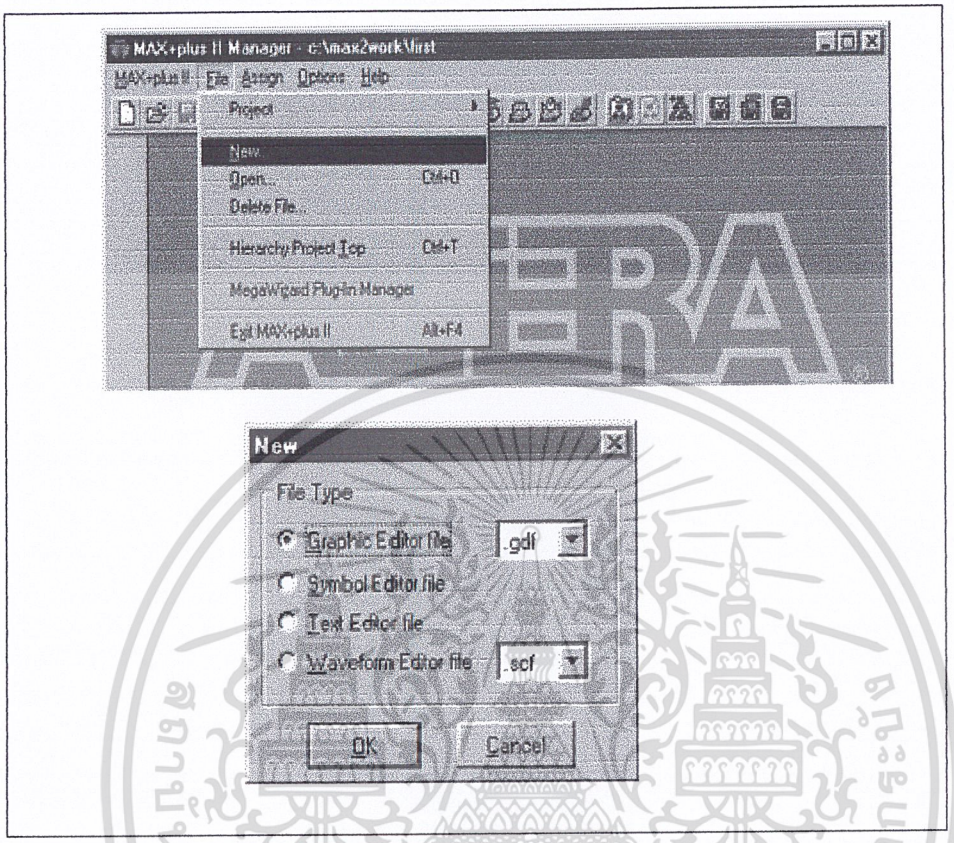
2. ทำการสร้างโปรเจ็คสำหรับวงจรที่ต้องการออกแบบโดยเริ่มจากการตั้งชื่อของโปรเจ็คที่เมนู File/Project / Name สำหรับตัวอย่างนี้ให้ตั้งชื่อโปรเจ็คเป็น First



รูปที่ 5.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เลือกประเภทของไฟล์ที่จะสร้างจากเมนู File / New / Graphic Editor File จากนั้นจะมีไดอะล็อก ให้เลือกประเภทของไฟล์ที่ต้องการสร้างขึ้นมา



รูปที่ 5.4

ไฟล์แต่ละชนิดจะมีความหมายดังต่อไปนี้

Graphic Editor File : เป็นไฟล์กราฟิกที่เราสามารถนำอุปกรณ์ต่างๆ ใน Library มาวางต่อกันเป็นวงจรได้เลย

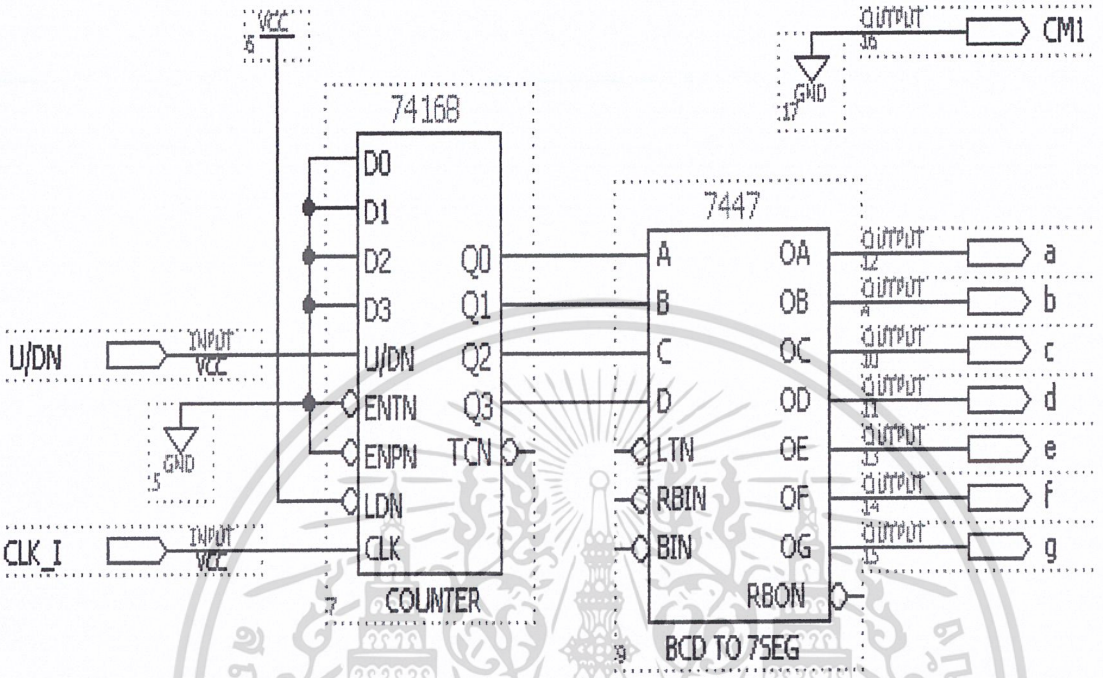
Symbol Editor File : เป็นไฟล์ที่ใช้เก็บสัญลักษณ์เพื่อสื่อให้ทราบว่าโมเดลที่เราได้สร้างขึ้นมีอินพุทและเอาท์พุทเป็นอย่างไร

Text Editor File : เป็น Text file ใช้สำหรับเขียน Source code เพื่ออธิบาย พฤติกรรมของวงจรหรือโมเดลต่างๆ ที่เราจะสร้างหรือเพื่อใช้สำหรับเก็บข้อความทั่วๆไป

Waveform Editor File : เป็นไฟล์ Waveform ที่มีไว้สำหรับอธิบายลักษณะ Waveform ของวงจรที่เราต้องการ ซึ่งถ้ามีการกำหนด Waveform ทั้งทางอินพุทและเอาท์พุทก็จะใช้นามสกุลของไฟล์เป็น wdf (Waveform Design File) แต่ถ้าใช้ในการกำหนดรูปแบบของสัญญาณอินพุท เพื่อให้โปรแกรมจำลองการทำงานและบันทึกผลที่ได้จะใช้ไฟล์ที่มีนามสกุลเป็น scf (Simulate Channel File)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ในการอธิบายการใช้งานโปรแกรม MAXPlus II ในบทนี้เราจะใช้วงจรในรูปที่ 5.5 ประกอบกาอธิบาย

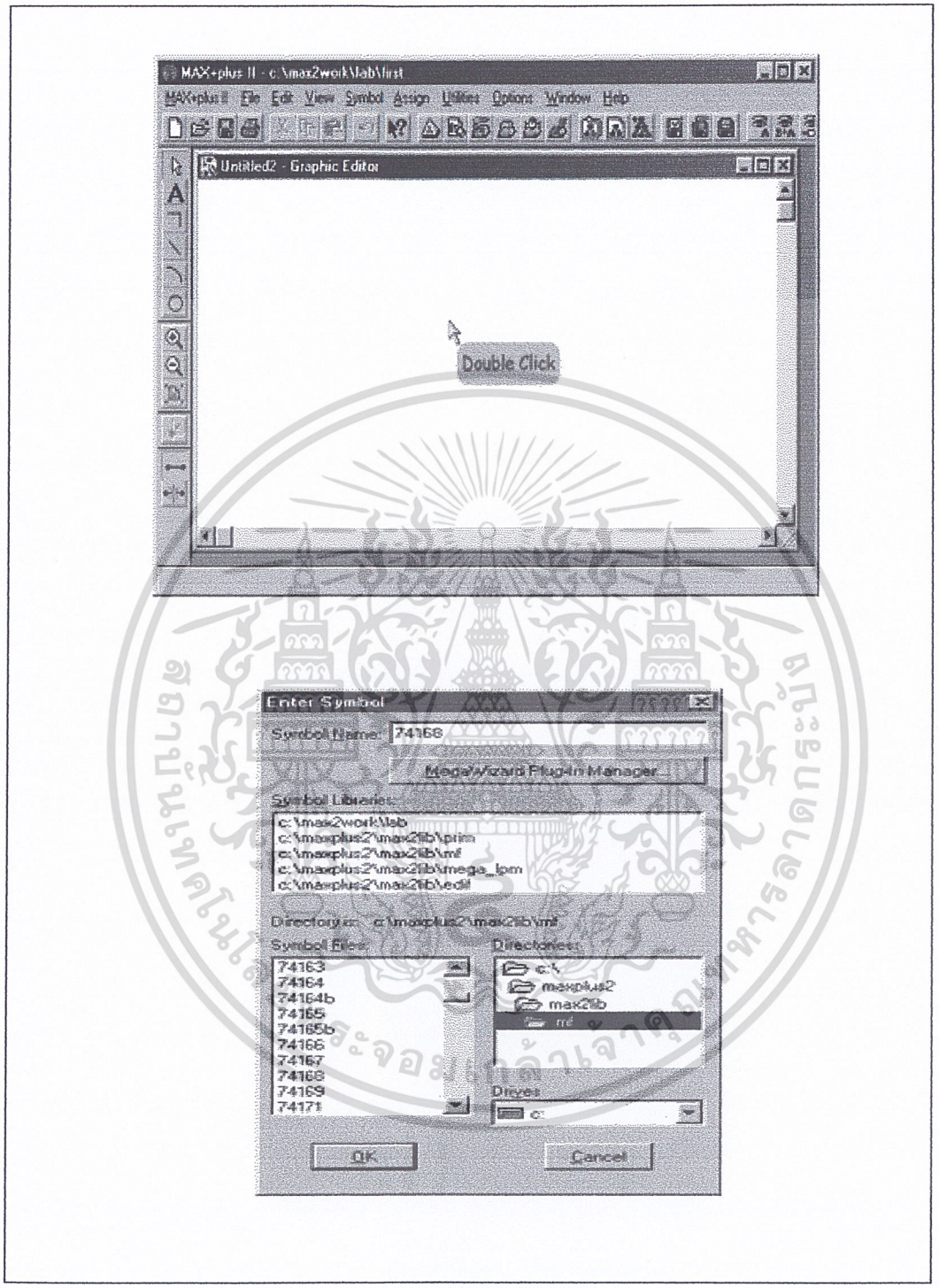


รูปที่ 5.5

จากรูปที่ 5.5 เป็นวงจร Decade Counter ซึ่งมีอุปกรณ์ที่ใช้ได้แก่ 74168, 7447, INPUT, OUTPUT, VCC, GND โดยอุปกรณ์ต่างๆ เหล่านี้จะมาพร้อมกับโปรแกรม MAX+PLUSE II ในรูปของไลบรารีส่วนการ นำอุปกรณ์ต่างๆ เหล่านี้มาใช้ให้ทำการดับเบิ้ลคลิกที่ Graphic Form ดังรูปที่ 5.6 และหลังจากนั้นจะมี Dialog Enter Symbol ปรากฏขึ้นมาเพื่อให้เลือกอุปกรณ์ที่เราต้องการนำมาวางที่ Graphic Form โดยที่

- 7447 ใช้ชื่อ Symbol Name ว่า 7447
- INPUT ใช้ชื่อ Symbol Name ว่า input
- OUTPUT ชื่อ Symbol Name ว่า output
- VCC ใช้ชื่อ Symbol Name ว่า vcc
- GND ใช้ชื่อ Symbol Name ว่า gnd
- 74168 ใช้ชื่อ Symbol Name ว่า 74168

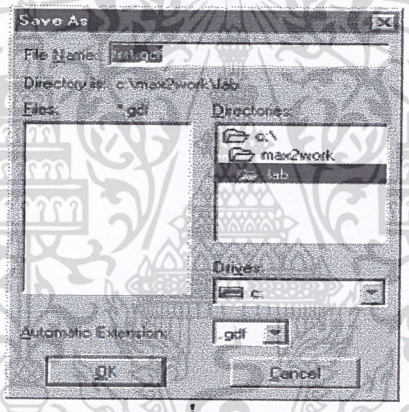
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

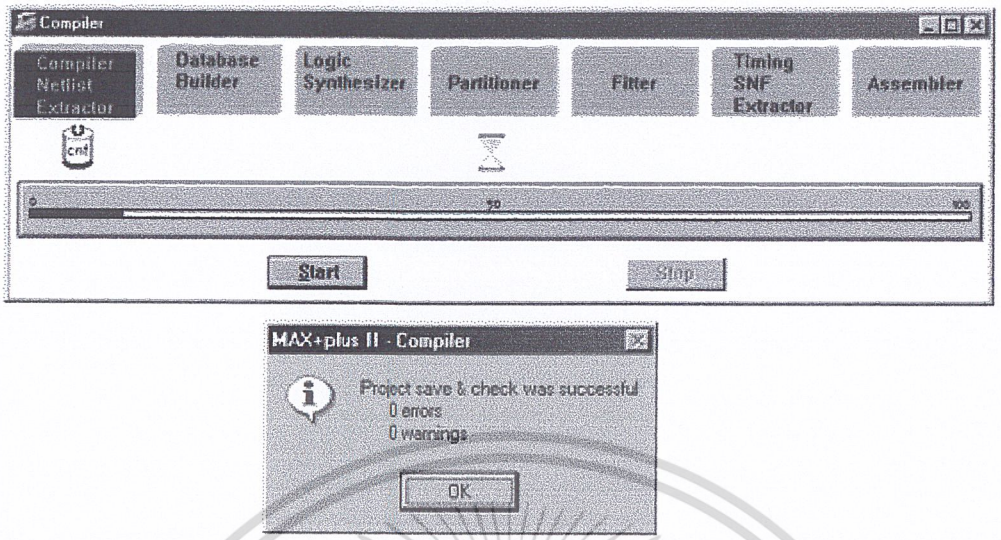
สำหรับอุปกรณ์ 74168 และ 7447 จะอยู่ใน Symbol Library ...\maxplus2 \max2lib\mf ส่วน INPUT, OUTPUT, VCC, GND จะอยู่ใน Symbol Library ...\maxplus2\max2lib\prim เมื่อทำการวาง อุปกรณ์จนครบทุกตัวแล้ว ให้ทำการเชื่อมต่ออุปกรณ์ต่างๆ เข้าด้วยกันโดยเลื่อน pointer ของเมาส์ไปที่ขาต่างๆ ของอุปกรณ์ คุณจะสังเกตได้ว่าขณะนี้ pointer ของเมาส์จะเปลี่ยนเป็นเครื่องหมาย + จากนั้นให้ทำการคลิกซ้าย ค้างไว้แล้วลากเส้น ไปต่อกับอินพุทหรือเอาท์พุทของอุปกรณ์ตัวอื่น และถ้าหากต้องการลบเส้นที่วาดไว้ ให้คลิกที่เส้น ที่ต้องการลบซึ่งขณะนี้เส้นนั้นจะเปลี่ยนจากสีดำมาเป็นสีแดง จากนั้นกดปุ่ม Delete บนคีย์บอร์ดเพื่อลบเส้นนั้นทิ้ง หลังจากเชื่อมต่ออุปกรณ์ต่างๆ จนครบทั้งหมดแล้วให้ทำการเปลี่ยนชื่อ ของขาอินพุทและเอาท์พุทต่างๆ หรือ PIN_NAME ของอุปกรณ์โดยการดับเบิ้ลคลิกที่ตัวอักษร PIN_NAME ที่ขาอินพุทและเอาท์พุทต่างๆ แล้วเปลี่ยน ชื่อให้ตรงกับรูปที่ 5 เมื่อวาดรูป วงจรเสร็จเรียบร้อยแล้วให้ทำการบันทึกวงจรที่สร้างขึ้นมาจากเมนู File / Save as จากนั้น Save ชื่อไฟล์เป็น first.gdf แล้วคลิกที่ปุ่ม OK



รูปที่ 5.7

5. ตรวจสอบความถูกต้องในการต่อวงจร File / Project / Save& Check หากต่อวงจรถูกต้องตามเงื่อนไข ของโปรแกรม MAXPlus II จะมีหน้าต่างปรากฏขึ้นมาดังรูป 5.8

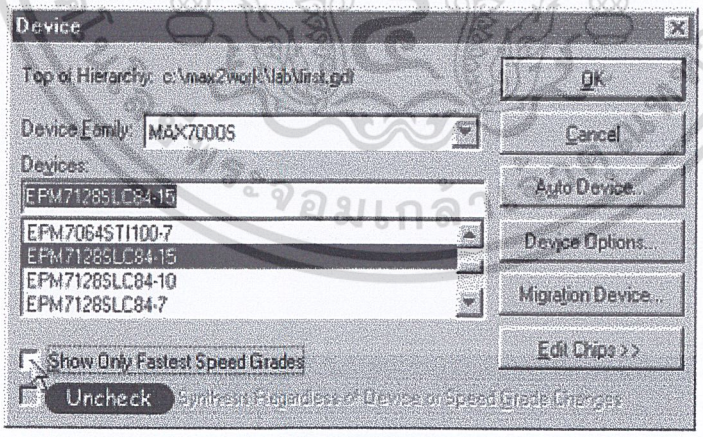
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.8

การตรวจสอบและสังเคราะห์วงจร

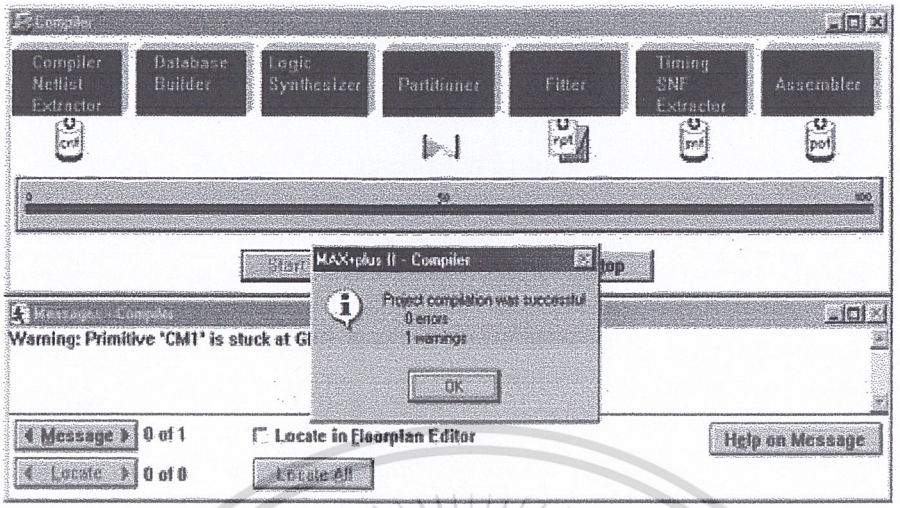
6. ขั้นตอนนี้เป็นการตรวจสอบและสังเคราะห์วงจร (Compiler and Synthesis) เริ่มจากทำการระบุเบอร์ของ ชิพที่จะใช้จากเมนู Assign / Device ซึ่งจะมีไดอะล็อก Device ปรากฏขึ้นมาเพื่อให้เลือก Device ที่ต้องการ จากนั้นทำการเลือก Device Family เป็น MAX7000S และเลือก Device เป็น EPM7128SLC84-15 ซึ่งต้อง Uncheck ช่อง Show Only Fastest Speed Grades ก่อนเพื่อให้ไดอะล็อกแสดง Device ที่มีอยู่ ทั้งหมดขึ้นมาดังรูปที่ 5.9



รูปที่ 5.9

7. ทำการคอมไพล์วงจรที่ได้สร้างขึ้นจากเมนู MAX+PLUS II / Compiler หลังจากนั้นจะมีไดอะล็อก Compiler ปรากฏขึ้นมา ให้กดปุ่ม Start เพื่อเริ่มทำการคอมไพล์หลังจากการคอมไพล์เสร็จสิ้นลงก็จะมี หน้าต่างรายงานผลการคอมไพล์ error และ warning หากมีข้อผิดพลาดเกิดขึ้นก็จะมีข้อความสีแดงบอก ว่า error พร้อมทั้งสาเหตุของข้อผิดพลาด

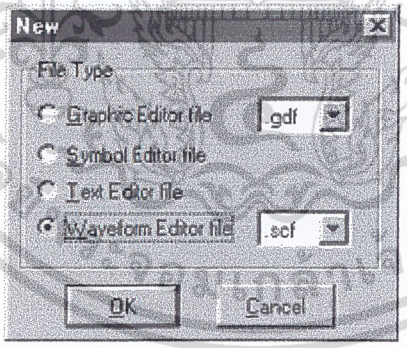
ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่ ห้ามนำไปใช้ด้านการค้า ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.10

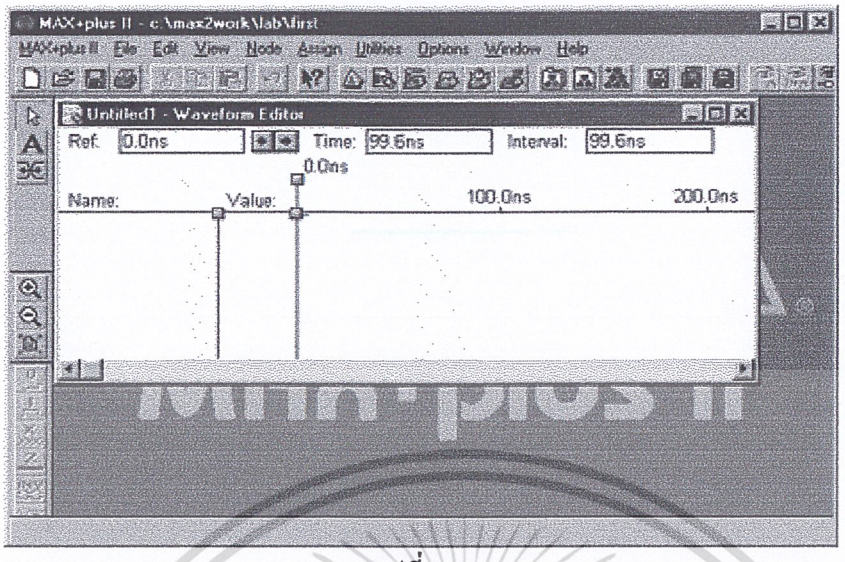
จำลองการทำงาน (Simulation)

8. จำลองการทำงานของวงจร (Simulation) โดยสร้างไฟล์ Waveform จากเมนู File / New / Waveform Editor file ซึ่งจะมีไดอะล็อกปรากฏขึ้นดังรูปที่ 5.11 โดยในขณะนี้ Wave Editor จะปรากฏขึ้นมาดังรูปที่ 5.12



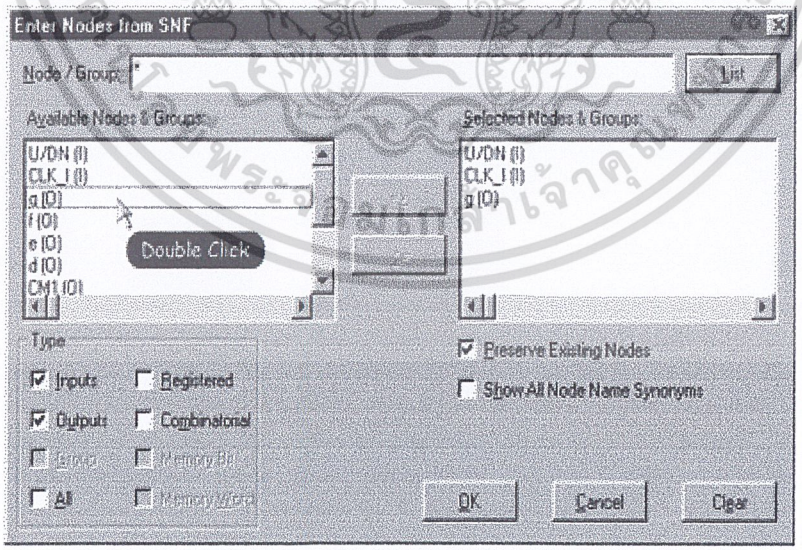
รูปที่ 5.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.12

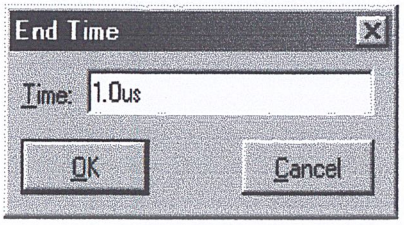
ก่อนทำการ Simulate เราจะต้องกำหนดลักษณะของสัญญาณอินพุทให้แก่วงจรก่อน โดยการโหลดโหนด ต่างๆ เข้ามาโดยใช้เมนู Noad / Enter Nodes from SNF... ซึ่งจะมีไดอะล็อก Enter Nodes from SNF ปรากฏขึ้นมา ต่อไปให้คลิกที่ปุ่ม List เพื่อแสดงโหนดต่างๆ ที่อยู่ในวงจรขึ้นมา จากนั้นเลือกโหนดที่เป็น input และ output ทั้งหมดที่อยู่ใน Listboทางด้านซ้ายโดยการดับเบิ้ลคลิกทุกๆ รายการที่อยู่ในช่องนี้ คุณจะพบว่าในรายการต่างๆ จะเข้าไปอยู่ใน Listbox ทางด้านขวาหลังจากการดับเบิลคลิก ดังรูปที่ 5.13



รูปที่ 5.13

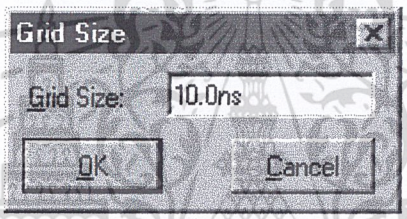
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. กำหนดเวลาสิ้นสุดการ Simulate (End Time) ให้กับโปรแกรมจากเมนู File / End Time... จากนั้น จะมีไดอะล็อก End Time ปรากฏขึ้นมา ให้ใส่ค่า End Time เท่ากับ 1.0 us ซึ่งจะเป็นการบอกให้โปรแกรม ทำการ Simulate ตั้งแต่ 0.0 us จนถึง 1.0 us ดังรูปที่ 5.14



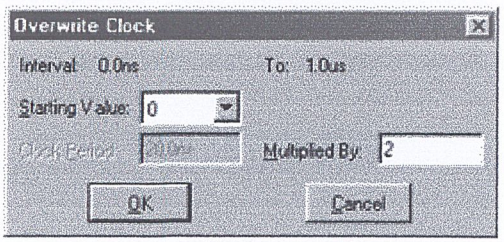
รูปที่ 5.14

10. กำหนดขนาดของกริดจากเมนู Option / Grid Size.. ซึ่งจะมีไดอะล็อก Grid Size ปรากฏขึ้นมา จากนั้นกำหนดให้กริดมีขนาดเท่ากับ 10.0 ns ดังรูปที่ 5.15



รูปที่ 5.15

11. ทำการกำหนดรูปแบบสัญญาณให้กับ Node Input โดยการ คลิกที่ CLK_I ที่ Waveform Editor ให้มีแถบดำปรากฏขึ้นมาหลังจากนั้นทำการกำหนดรูปแบบสัญญาณให้มีลักษณะเป็นพัลซ์โดยใช้เมนู Edit / Overwrite / Clock โดยจะมีไดอะล็อก Overwrite Clock ปรากฏขึ้น ในช่อง Multiply By ให้ใส่ 2 แล้วกดที่ปุ่ม OK ซึ่งจะเป็นการกำหนดให้สัญญาณในช่วงลอจิก "1" และลอจิก "0" มีค่าเวลาเป็น 2 เท่าของ Grid Size

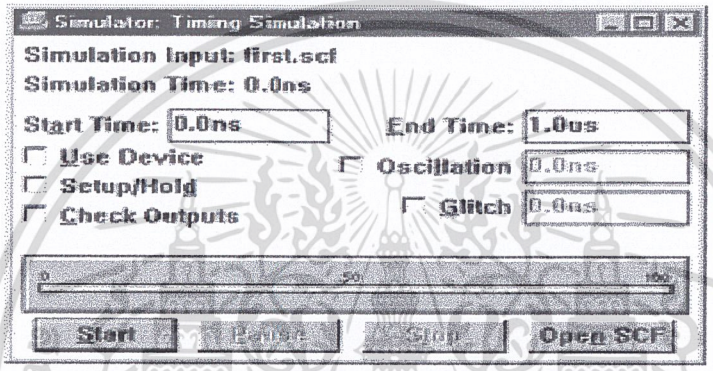


รูปที่ 5.16

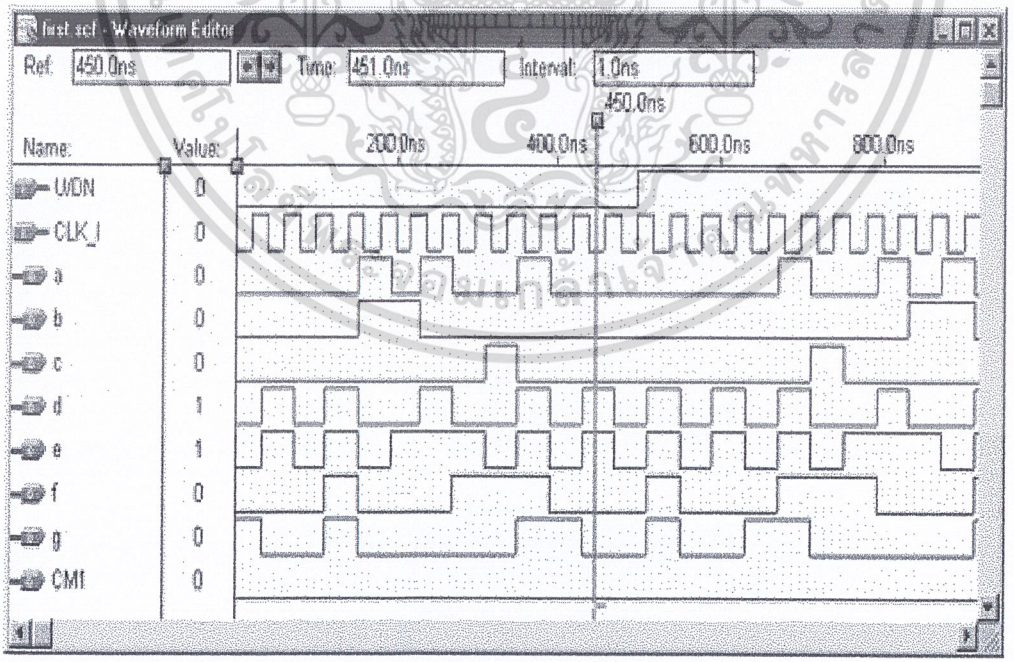
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของ U/DN ให้กำหนดรูปแบบสัญญาณเช่นเดียวกับ CLK_I ส่วนในช่อง Multiply By ให้มีค่าเท่ากับ 50 หลังจากนั้นบันทึกไฟล์ Waveform ที่ได้สร้างขึ้น File / Save as กำหนดให้ชื่อไฟล์ที่จะบันทึกเป็น First.scf

12. จำลองการทำงานของวงจรจากเมนู MAX+PLUS II / Simulator ซึ่งจะมีหน้าต่าง Timing Simulation ปรากฏขึ้นดังรูปที่ 5.17 มาจากนั้นให้กดปุ่ม Start เพื่อทำการ Simulate วงจร และเมื่อ การคำนวณเสร็จสิ้นลงก็จะมีไดอะล็อกขึ้นมารายงานผลการ Simulate ว่ามี error หรือ warning หรือไม่ ส่วนผลของการ Simulate จะปรากฏที่ Waveform Editor ดังรูปที่ 5.18



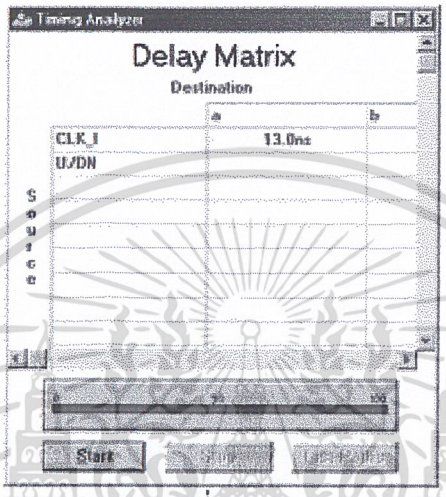
รูปที่ 5.17



รูปที่ 5.18

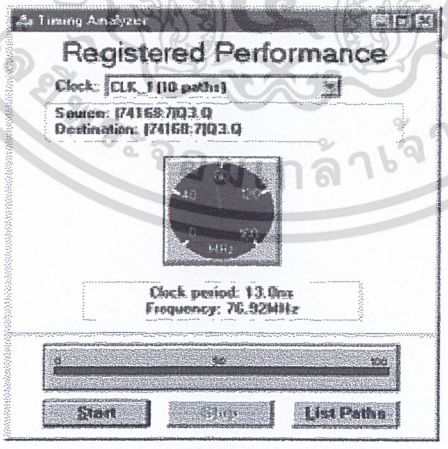
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. จากผลการจำลองการทำงาน สามารถนำมาวิเคราะห์ Timing เพื่อใช้สำหรับหาค่า Delay Time ระหว่างโหนดต่างๆ โดยใช้เมนู MAXPlus II / Timing Analyzer ซึ่งจะมีไดอะล็อก Timing Analyzer ปรากฏขึ้นมา จากนั้นทำการวิเคราะห์ค่าเวลาหน่วงจากเมนู Analysis / Delay Time ดังรูปที่ 5.19



รูปที่ 5.19

นอกจากนี้ยังสามารถวิเคราะห์ คำนวณหาค่าความถี่สูงสุดที่วงจรสามารถทำงานได้ โดยการเลือกที่เมนู Analysis / Registered Performance ดังรูปที่ 5.20

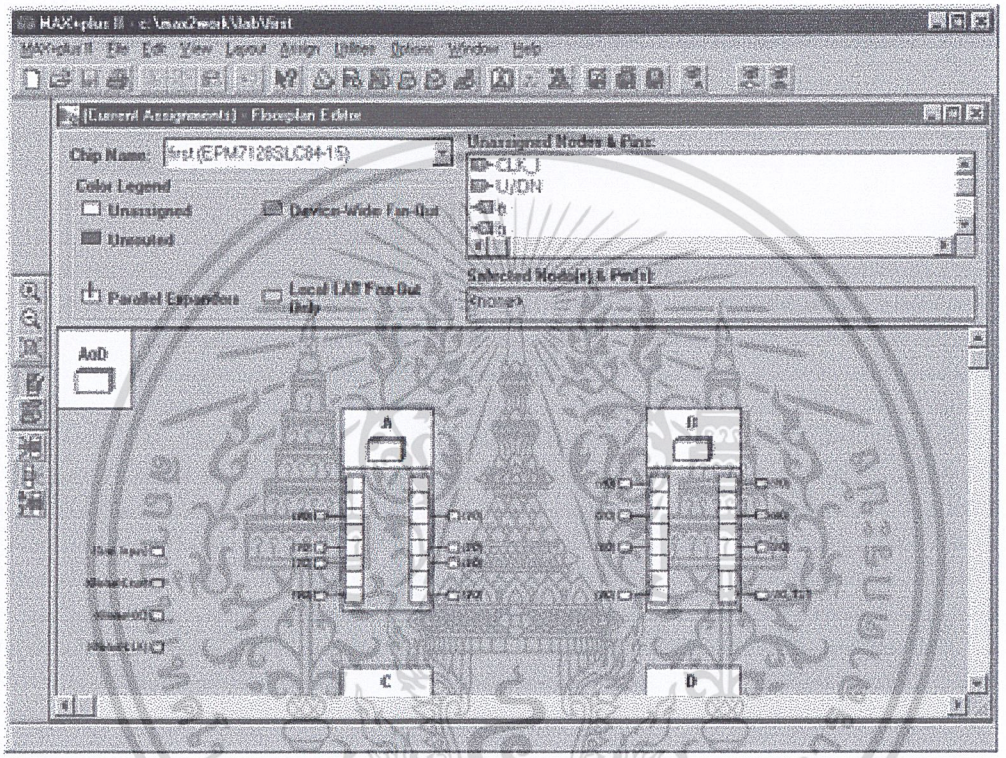


รูปที่ 5.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การโปรแกรมลงชิพ FPGA

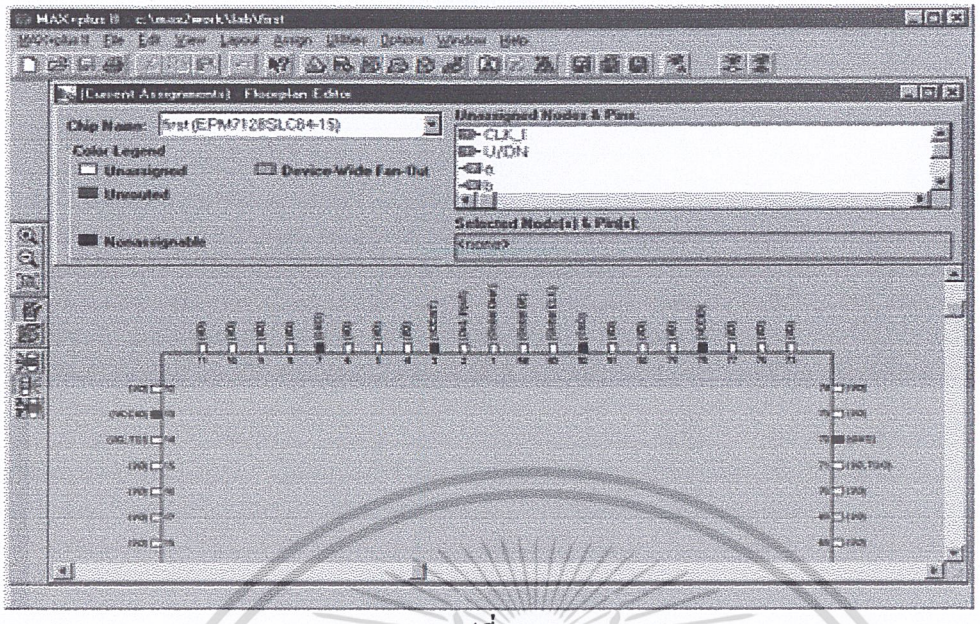
14. ขั้นตอนนี้จะเป็นการโปรแกรมวงจร ที่ออกแบบไว้ลงในชิพ FPGA ซึ่งก่อนที่จะทำการโปรแกรมนั้นจำเป็นต้อง มีการกำหนดตำแหน่งขาของอุปกรณ์ต่างๆ ที่อยู่ในวงจรกับขาของ FPGA ก่อน โดยเริ่มจากการเรียกเมนู Max Plus / Floorplan Editor หลังจากนั้นหน้าต่าง Floorplan Editor จะปรากฏขึ้นมาดังรูปที่ 5.21 ซึ่งเป็น Layout แบบ LAB View (Logic Array Block View)



รูปที่ 5.21

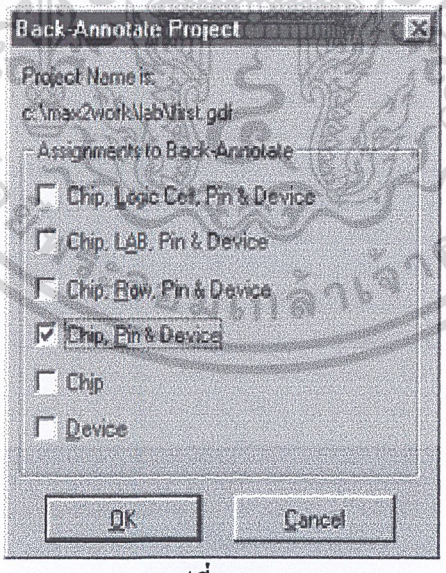
เนื่องจาก Layout แบบ LAB View นี้ อาจพิจารณาได้ลำบากในแง่ของการต่อวงจร ดังนั้นจึงควรเปลี่ยน Layout ให้เป็นแบบ Device View ดังรูปที่ 5.22 ซึ่ง Layout แบบนี้จะเป็นการมองจากตำแหน่งขาที่แท้จริงของ FPGA เบอร์ที่เราระบุไว้โดยการเรียกเมนู Layout / Device View

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.22

หลังจากนั้นทำการกำหนดกำหนดอินพุทและเอาท์พุทของวงจรลงไปโดยเรียกเมนู Assign / Back-Annotate Project... ซึ่งจะมีไดอะล็อก Back-Annotate Project ปรากฏขึ้นมาให้คลิกที่ Chip, Pin & Device / OKดังรูปที่ 5.23



รูปที่ 5.23

สังเกตได้ว่าเมื่อผ่านขั้นตอนนี้ไปแล้วโปรแกรม MAXPlus II จะทำการกำหนดตำแหน่งขาต่างๆ มาให้เรียบร้อย แล้วแต่อาจยังไม่เหมาะสม กับการเชื่อมต่อของบอร์ดทดลอง ดังนั้นจึงต้องทำการเปลี่ยนแปลงตำแหน่งขาต่าง ๆ ของ ชิพ FPGA ก่อนโดยเลือกเมนู Layout / Current Assignment Floorplan

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปลี่ยนแปลงตำแหน่งขาของชิพ FPGA สามารถทำได้โดยการคลิกซ้ายที่ขาของชิพค้างไว้ แล้วเคลื่อนไปวางไว้ที่ตำแหน่งใหม่ โดยที่

CLK_I ต่อกับขาที่ 83

U/DN ต่อกับขาที่ 49

a ต่อกับขาที่ 33

b ต่อกับขาที่ 34

c ต่อกับขาที่ 35

d ต่อกับขาที่ 36

e ต่อกับขาที่ 37

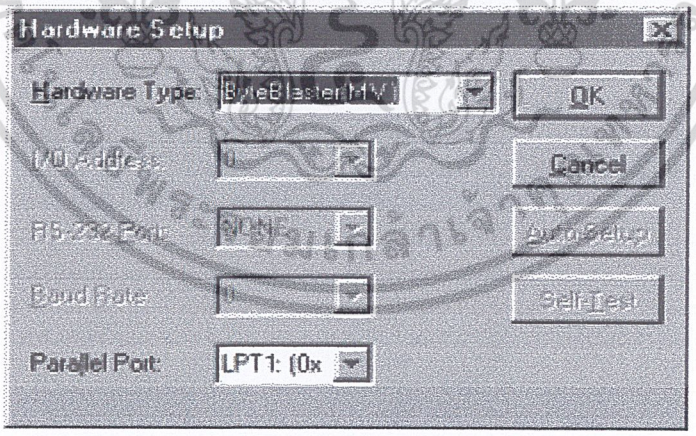
f ต่อกับขาที่ 39

g ต่อกับขาที่ 40

CM1 ต่อกับขาที่ 44

หลังจากทำการเปลี่ยนแปลงตำแหน่งขาของชิพเรียบร้อยแล้วให้ Save และ คอมไพล์ใหม่อีกครั้ง

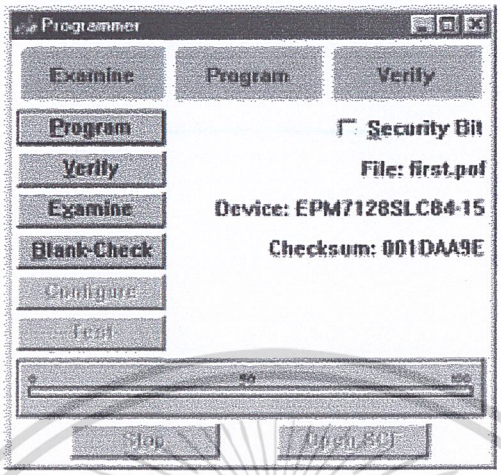
15.ทำการโปรแกรมวงจรลงในชิพ FPGA โดยเลือกที่เมนู MAXPuls II / Programmer ซึ่งหากยังไม่ เคยติดตั้งอุปกรณ์สำหรับโปรแกรมชิพ FPGA (สาย ByteBlaster) ก็จะมีไดอะล็อก Hardware Setup ปรากฏขึ้นมา ให้นำสาย Byte Blaster มาต่อที่พอร์ตขนาน (พอร์ต Printer) แล้วเลือก Hardware Type เป็น ByteBlaster (MV) และกดปุ่ม OK ดังรูปที่ 5.24



รูปที่ 5.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อโปรแกรม MAX+PLUS II ตรวจพบว่ามีสาย Byte Blaster ต่ออยู่ที่พอร์ตขนานแล้ว หน้าต่าง Programmer จะเป็นดังรูปที่ 5.25 จากนั้นให้กดปุ่ม Program เพื่อโหลดวงจรลงชิพ FPGA



รูปที่ 5.25

ข้อสังเกตอย่างหนึ่งในการโปรแกรมวงจรคือหาก Device ของ FPGA ที่เลือกเป็นอุปกรณ์ประเภท EEPROM- Base FPGA หรือที่เรียกว่า CPLD (เบอร์ IC จะขึ้นต้นด้วย EPM) ไฟล์ที่โปรแกรมจะมีนามสกุลเป็น pof ซึ่งก่อน ที่จะโปรแกรมวงจรลงในชิพ FPGA ให้ทำการตรวจสอบก่อนว่าได้ทำการต่อสาย ByteBlaster กับบอร์ดทดลอง เรียบร้อยแล้ว ได้ทำการจ่ายไฟให้กับบอร์ดทอลองแล้ว มิฉะนั้นการโปรแกรมข้อมูลลงในชิพ FPGA จะเกิดการผิด พลาดได้

ทดสอบการทำงานของวงจร

16. จากวงจรตัวอย่างเป็นวงจรนับ (Counter) จะทำการนับตั้งแต่ 0-9 ในขณะที่ U/DN เป็น Logic "1" จะนับขึ้น และหาก U/DN เปลี่ยนเป็น Logic "0" ก็จะนับลง แต่เนื่องจากสัญญาณ Clock ในบอร์ดทดลองมีความถี่สูงมาก ดังนั้น เมื่อทำการโปรแกรมลงไปแล้ว จะสังเกตผลที่ 7-Segment ได้ไม่ชัดเจน ซึ่งการแก้ไขอาจทำได้โดยการนำวงจรหาร มาทำ การหารความถี่ของสัญญาณ Clock ลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การออกแบบและการสร้าง

ในส่วนของการออกแบบและการสร้างได้แบ่งออกเป็นสองส่วนคือ

- การออกแบบส่วนของฮาร์ด ซึ่งประกอบไปด้วย พร๊อตต่อจอ VGA วงจรจ่ายแรงดัน อุปกรณ์ FPGA ไดโอด สวิช์ทกดติดปล่อยดับ IC#THS8134B รีเลย์ 6V ตัวออสซิเลเตอร์ แบตเตอร์รี่ 3.6 V และก็ R และ C

- การออกแบบในส่วนของซอร์ฟแวร์ เป็นการออกแบบการให้กำเนิดสัญญาณที่ใช้ควบคุมจอภาพ VGA โดยจะมีการแสดงผลตามสวิช์ทที่กด

6.1 การออกแบบส่วนฮาร์ดแวร์

1. วงจรจ่ายแรงดันคงที่

เป็นการใช้ ไอซีเรกูเลเตอร์ เบอร์ 7805 เป็นตัวทำให้ได้แรงดันคงที่ 5 VDC ซึ่งเป็นแรงดันที่จ่ายให้กับตัว FPGA และเนื่องจากตัว FPGA เบอร์นี้จะต้องมีไฟเลี้ยงอยู่ตลอดเวลา โปรแกรมที่ทำการโปรแกรมไว้จึงจะไม่หายไปจึงมีการเพิ่มแบตเตอร์รี่ เพื่อเป็นตัว back up ข้อมูลไว้โดยใช้รีเลย์เป็นตัวตัดต่อ

2. ออสซิลเลเตอร์ (Oscillator)

ใช้ออสซิลเลเตอร์กำเนิดความถี่ 25.175 MHz ให้กับตัว FPGA และ IC#THS8134B เปรียบเสมือนกับเป็นตัวควบคุมการทำงานของ อุปกรณ์ต่างๆ

3. ตัวขับจอ VGA

ในภาคนี้เราจะใช้ IC#THS8134B เป็นตัวที่แปลงสัญญาณดิจิตอลให้เป็นสัญญาณอนาลอกเพื่อใช้ในการควบคุมจอ VGA

4. การโหลดข้อมูล

การโหลดข้อมูลทำได้โดยการใช้สาย Byte Blaster เป็นสายที่ใช้ในการดาว์นโหลดข้อมูลทางลอจิกที่จะทำการโปรแกรม ลงบนอุปกรณ์ FPGA ซึ่งทำการดาว์นโหลดโดยใช้โปรแกรม MAX Plus II ผ่านทางพล๊อตปริ๊นเตอร์ของคอมพิวเตอร์

5. สวิตช์กดติดปล่อยดับ และ ไดโอดเปล่งแสง (LED)

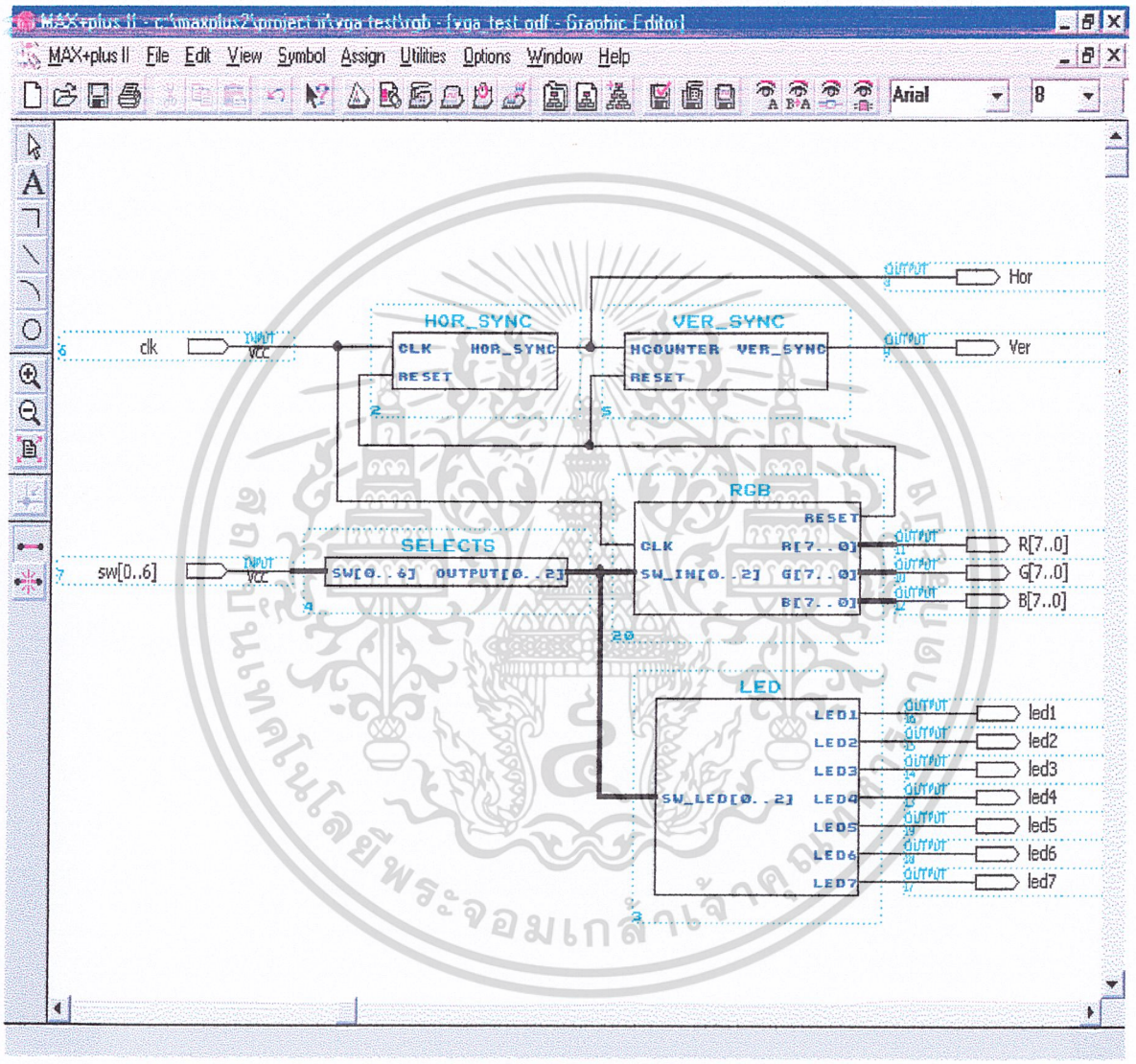
สวิตช์พวกนี้จะเป็นตัวใช้เลือกว่าจะให้เอาท์พุทออกมาในรูปใด โดยจะแสดงว่าสวิตช์ตัวนั้นถูกกดโดยตัว ไดโอดเปล่งแสง (LED) จะเปล่งแสงออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.1 วงจรอิเล็กทรอนิกส์ที่ใช้ในการทดลอง

6.2 การออกแบบส่วนซอร์ฟแวร์

เป็นการออกแบบการควบคุมการทำงานของอุปกรณ์ต่างๆ การเขียนโปรแกรมได้ใช้โปรแกรม MAX Plus II ซึ่งเป็นโปรแกรมสำเร็จรูปของ บริษัท Altera โดยการใช้ภาษา VHDL และแผนภาพวงจร (Schematics)



รูปที่6.2 แสดงภาพวงจรที่ออกแบบโดยใช้ MAX Plus II

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

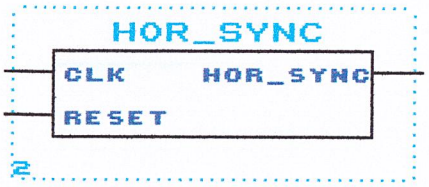
จากรูปที่ 6.2 สามารถเขียนเป็น Block Diagram ของขั้นตอนการทำงานได้ดังนี้



รูปที่ 6.3 แสดง Block Diagram ขั้นตอนการทำงาน

จากรูปที่ 6.2 สามารถอธิบายการทำงานของบล็อกต่างๆดังนี้

1. HOR_SYNC



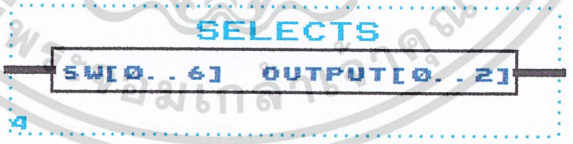
เป็นบล็อกที่ทำหน้าที่ให้กำเนิดสัญญาณการสแกนภาพทางแนวนอน (Horizontal Sync) โดยการนับจำนวน clock ที่เข้ามาเพื่อที่จะทำการกำเนิดสัญญาณ

2. VER_SYNC



เป็นบล็อกที่ทำหน้าที่ให้กำเนิดสัญญาณการสแกนภาพทางแนวตั้ง (Vertical Sync) โดยจะทำการนับจำนวนของเส้นสแกนแนวนอนเพื่อที่จะทำการกำเนิดสัญญาณ

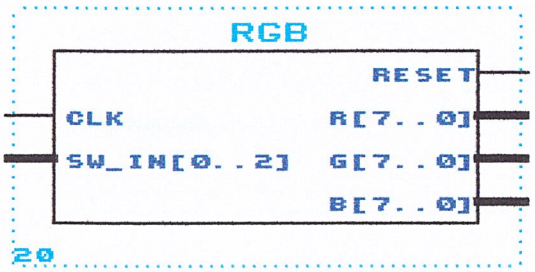
3. SELECTS



เป็นตัวทำหน้าที่เลือกสัญญาณเอาท์พุต RGB ว่าต้องการให้ออกมาในรูปแบบไหน โดยจะมีจำนวนตัวเลือก สวิช์ทอยู่ 7 อินพุต

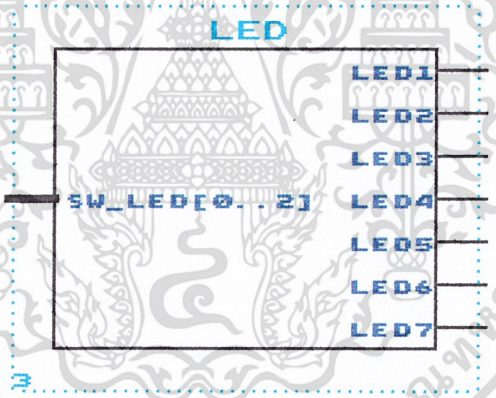
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. RGB



บล็อกนี้เป็นบล็อกที่เป็นส่วนแสดงผลออกเอาท์พุต โดยจะเป็นบล็อกที่เป็นตัวกำเนิดสัญญาณสีของ RGB ออกมาในรูปแบบดิจิตอลก่อนที่จะผ่านตัว D to A ออกไปขับจอ VGA ต่อไป โดยสัญญาณที่ให้กำเนิดออกมาจะมีอยู่ด้วยกัน 7 แบบ โดยเอาท์พุตที่ออกมาจะถูกเลือกด้วย สวิทช์ต่างๆ

5. LED



ส่วนในบล็อกสุดท้ายนี้เป็นส่วนที่จะแสดงผลออก LED เพื่อที่จะแสดงให้ทราบว่าสวิทช์ตัวไหนถูกกด และกำลังทำงานอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

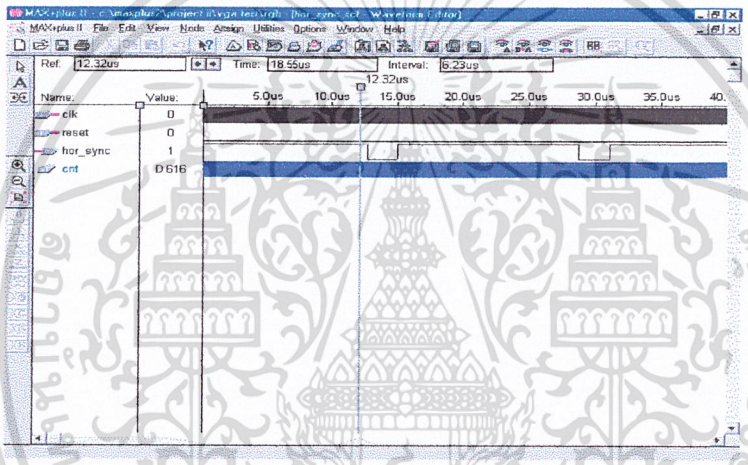
บทที่ 7

การทดลองและผลการทดลอง

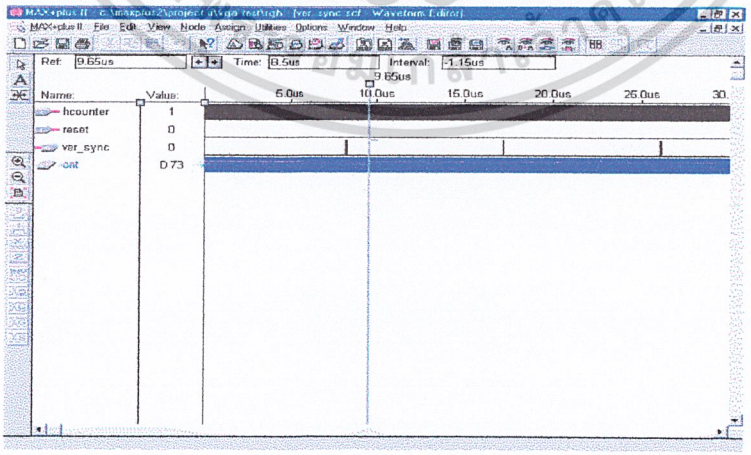
การทดลองและผลการทดลองได้แบ่งออกเป็นสองส่วนใหญ่ๆ

- การทดลองและผลการจำลองการทำงานในโปรแกรม MAXPluse II ซึ่งเป็นการทดสอบเกี่ยวกับการทำงานของโปรแกรมว่าถูกต้องหรือไม่อย่างไร ก่อนที่จะนำไป ทำการโหลดข้อมูลลงชิพ FPGA เพื่อทดสอบการทำงานจริงต่อไป
- ผลการทดลองของการทำงานจริงๆ ซึ่งเป็นการทดลองกับจอ VGA

7.1 ผลการทดลองการทำงานในโปรแกรม MAXPuls II

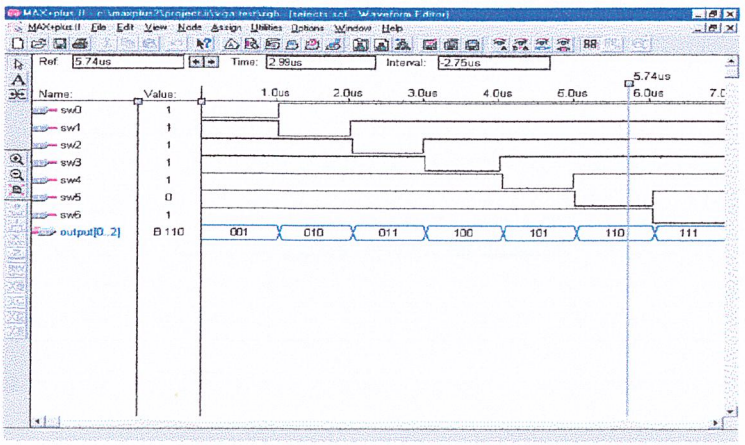


a) แสดงผลการจำลองการทำงานของบล็อก HOR_SYNC

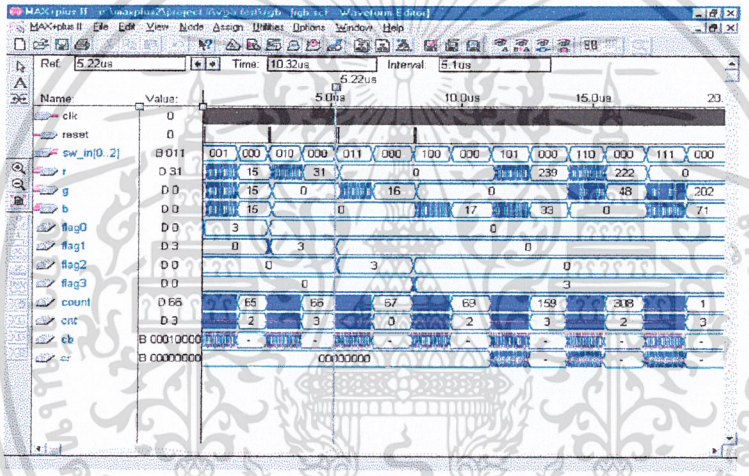


b) แสดงผลการจำลองการทำงานของบล็อกVER_SYNC

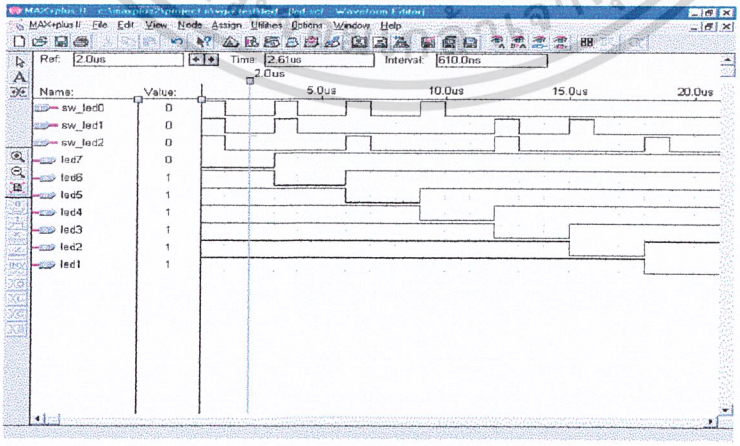
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



c) แสดงผลการจำลองการทำงานของบล็อก SELECT



d) แสดงผลการจำลองการทำงานของบล็อก RGB



e) แสดงผลการจำลองการทำงานของบล็อก LED

รูปที่ 7.1 ผลการ SIMULATE การทำงานของแต่ละบล็อก

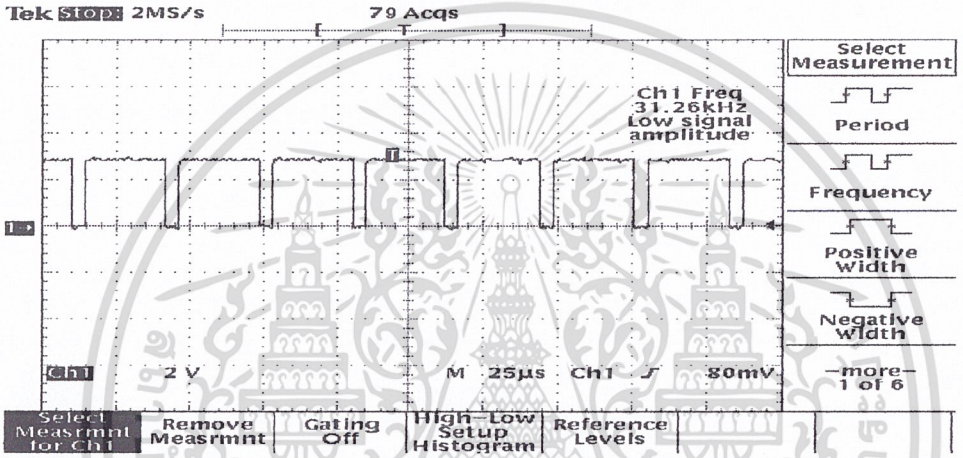
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 การทดลองผลการทดลองการทำงานจริง

ขั้นตอนการทดลอง

1. นำพล๊อตอินพุตของจอ VGA มาต่อกับ พล๊อต VGA ของบอร์ดทดลอง
2. ป้อนแหล่งจ่ายไฟ 5V ให้แก่บอร์ดทดลอง
3. ทำการโหลดโปรแกรมจากเครื่องคอมพิวเตอร์เข้าสู่บอร์ดทดลอง
4. ทำการกดสวิท์ชแต่ละตัวเพื่อดูผลของเอาท์พุทที่ได้ทางจอ VGA

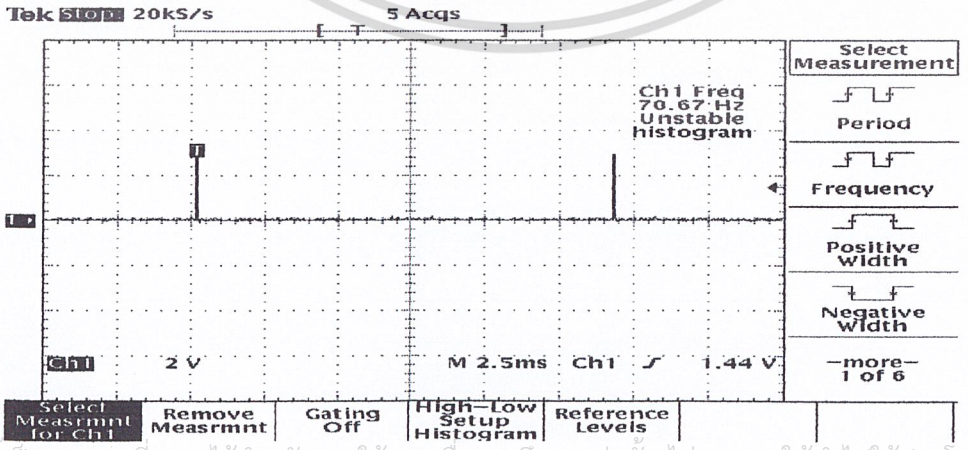
7.2.1 การทดลองวัดสัญญาณ HORIZONTAL_SYNC



รูปที่ 6.2 สัญญาณ HORRIZONTAL_SYNC

จะทำการวัดสัญญาณตรงขาที่ 11 ของ FPGA ซึ่งจะได้สัญญาณการสแกนทางแนวนอนใกล้เคียงกับในทฤษฎีจะมีค่าประมาณ 31 KHz

7.2.2 การทดลองวัดสัญญาณ VERTICAL_SYNC

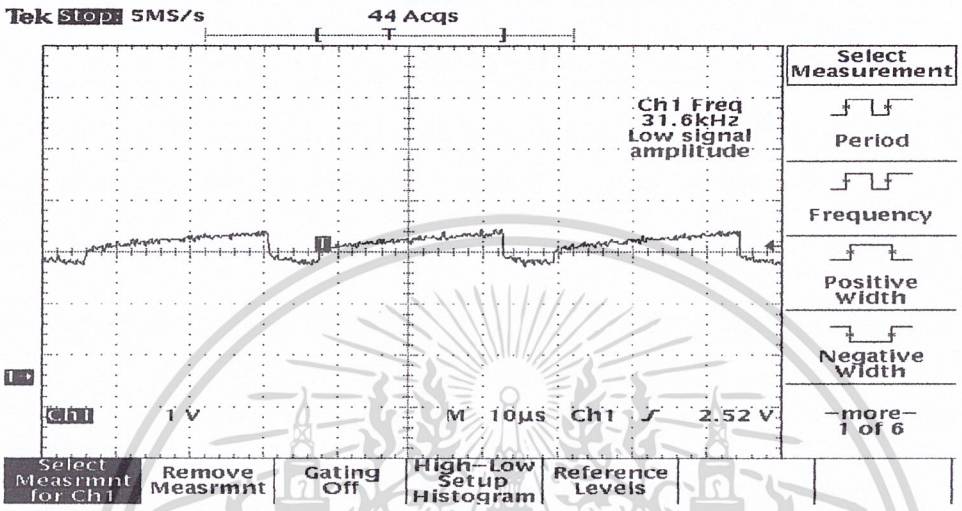


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.3 สัญญาณ VERTICAL_SYNC

จะทำการวัดสัญญาณตรงขาที่ 9 ของ FPGA ซึ่งจะได้สัญญาณการสแกนทางแนวตั้งใกล้เคียงกับในทฤษฎีจะมีค่าประมาณ 70 Hz

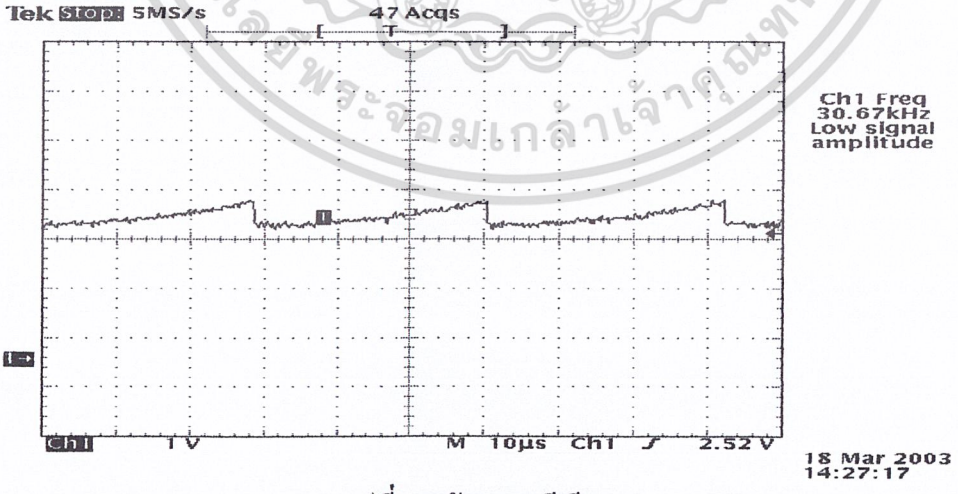
7.2.3 การทดลองวัดสัญญาณ R (สัญญาณสีแดง) เมื่อมีการกด SW1



รูปที่ 6.4 สัญญาณสีแดง

วัดสัญญาณสีแดงนี้จะสามารถวัดได้ที่ขา 43 ของ IC#THS8134B ซึ่งสัญญาณจะมีระดับการเพิ่มของแรงดันทำให้มีหลายระดับสีขึ้น

7.2.4 การทดลองวัดสัญญาณ G (สัญญาณสีเขียว) เมื่อมีการกด SW1

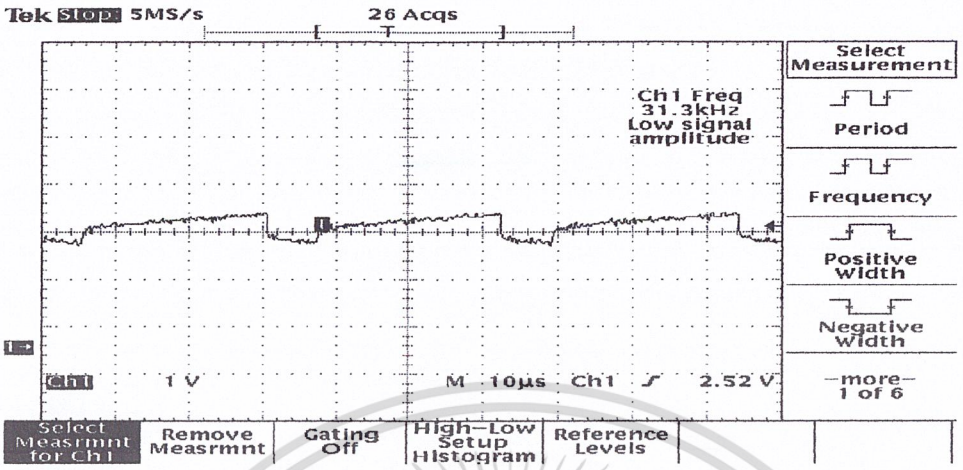


รูปที่ 6.4 สัญญาณสีเขียว

วัดสัญญาณสีเขียวนี้จะสามารถวัดได้ที่ขา 41 ของ IC#THS8134B ซึ่งสัญญาณจะมีระดับการเพิ่มของแรงดันทำให้มีหลายระดับสีขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2.5 การทดลองวัดสัญญาณ B (สัญญาณสีน้ำเงิน) เมื่อมีการกด SW1



รูปที่ 6.4 สัญญาณสีเขียว

วัดสัญญาณสีเขียวนี้จะสามารถวัดได้ที่ขา 45 ของ IC#THS8134B ซึ่งสัญญาณจะมีระดับการเพิ่มของแรงดันทำให้มีหลายระดับสีขึ้น

7.3 ผลการทดลอง

เมื่อทำการโหลดโปรแกรมจากเครื่องคอมพิวเตอร์ เข้าสู่บอร์ดทดลอง แล้วจะมีการทำงานของ FPGA โดยจะสังเกตได้จากจอ VGA นั้นจะทำงานแต่จะไม่มีภาพใดๆ ออกมาเพราะ ตัว FPGA ทำงานแค่ส่วนให้กำเนิดสัญญาณสแกนแนวนอนและสแกนแนวตั้งเท่านั้น และเมื่อมีการกดสวิทช์จะเป็นตัวสั่งให้ FPGA นั้นสร้างสัญญาณภาพออกมาโดยสวิทช์แต่ล่ะตัวจะสร้างสัญญาณแตกต่างกันไป คือเมื่อกดสวิทช์ 1 จะแสดงภาพที่มีการไล่ระดับสีของสีขาวและถ้ากดสวิทช์ 2 จะเป็นการแสดงภาพที่มีการไล่ระดับสีของสีแดง ส่วน สวิทช์ 3 จะแสดงภาพที่มีการไล่ระดับสีของสีเขียวและ สวิทช์ 4 จะแสดงภาพที่มีการไล่ระดับสีของสีน้ำเงิน สวิทช์ 5 นั้นเป็นการแสดงการไล่ระดับการผสมสีระหว่างสีแดงกับสีน้ำเงิน ส่วนในสวิทช์ 6 และ 7 ก็เป็นการไล่ระดับการผสมสีระหว่างสีแดงกับสีเขียว และ สีเขียวกับสีน้ำเงินตามลำดับ และเมื่อมีการกดสวิทช์แต่ละตัวจะมี LED คอยบอกอยู่ว่าสวิทช์ตัวใดถูกกดอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8
บทสรุป

8.1 สรุปผลการทดลอง

รายงานนี้ได้นำเสนอความรู้เกี่ยวกับ ทฤษฎี และการเขียน ภาษา VHDL , ความเป็นมาและเทคโนโลยีของ FPGA ว่าเป็นมาอย่างไร และการใช้งานต่างๆของ โปรแกรม MAXPlus II รวมไปถึงหลักการทำงานและขั้นตอนการทำงาน ของการสแกนจอภาพ

และจากการทดลองความถี่ของสัญญาณ Horizontal นั้นมีค่าอยู่ที่ 31 KHz และส่วนสัญญาณ Vertical นั้นจะมีคว่ามถี่อยู่ที่ 70 Hz และสัญญาณ RGB นั้นจะมีความถี่ที่เปลี่ยนแปลงไปมาตามเอาท์พุทที่ได้ออกมาตามที่ออกแบบไว้ด้วยภาษา VHDL

ส่วนแรงดันที่วัดได้ที่ขาของสัญญาณ RGB จะมีค่าอยู่ที่ 0V ถึง 1.5V ซึ่งเป็นระดับแรงดันที่เปลี่ยนแปลงไปมาเพื่อให้ได้ความระเอียดของระดับสีที่ได้ออกมา

8.2 ปัญหาและแนวทางการแก้ปัญหา

ตัวของ FPGA ที่ใช้มีเกตน้อยจึงสามารถทำรูปที่ใช้ในการทดสอบจึงจำกัด สามารถแก้โดยการเปลี่ยนมาใช้ชิพ FPGA ที่มีขนาดเกตมากขึ้นกว่านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้สำเร็จลุล่วงไปด้วยดีจากคำปรึกษาข้อเสนอแนะและห้องทำงานชั้น10 ที่ตึกกรมหลวงนราธิวาส จาก อาจารย์ที่ปรึกษา รศ.ดร.มนัส สังวรศิลป์ และขอขอบคุณ พี่ ศุภชัย ที่คอยให้คำแนะนำให้คำปรึกษาและคอยให้ความช่วยเหลือในทุกๆ ด้านด้วยดีตลอดมา รวมไปถึงพี่ๆ ปริญญาโททุกท่านที่ให้คำปรึกษาและให้ข้อมูลต่างๆที่เป็นประโยชน์ต่อการทำโครงงานและปริญญานิพนธ์เล่มนี้ ทางผู้จัดทำขอกราบขอบพระคุณเป็นอย่างยิ่ง

ศิรพงษ์ สัมฤทธิ์

(ศิรพงษ์ สัมฤทธิ์)

สุชาติ เก็งซี

(สุชาติ เก็งซี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

Stefan Sjoholm and Lennart Lindh , " VHDL for Designers " Pantice Hall

วิชรกร หนูทอง , " เอกสารประกอบการฝึกอบรม VHDL Application Workshop " ฝ่ายออกแบบวงจรรวมศูนย์วิจัยและพัฒนาเทคโนโลยีไมโครอิเล็กทรอนิกส์ (NECTEC) 2543

ทฤษฎี FPGA และ การใช้งาน MAXPlus II , www. Astronlogic.com

ชาติชาย ดิษฐกุล เอกสารประกอบการเรียน ภาษา VHDL , ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

Jame O. Hamblem, Michael D. Furman; "VGA VEDIO SIGNAL GENERATION" RAPID PHOTO TYPING OF DIGITAL SYSTEM, Geargia Institute of Technology , January 2000, pp. 133-152

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

โปรแกรมที่ใช้ในการกำเนิดสัญญาณ HORIZONTAL_SYNC

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
ENTITY hor_sync IS
        PORT (clk                       : IN  bit;
              reset                     : in  bit;
              hor_sync                  : OUT bit
             );
END ;
ARCHITECTURE hor_sync_a OF hor_sync IS
BEGIN
Process(clk)
Variable cnt : integer range 0 to 1023;
Begin
if clk'event and clk='1' then
 if reset = '1' then
        cnt:=0;
      elsif cnt<=640 then
              hor_sync<='1';
              cnt:=cnt+1;
      elsif cnt<=755 then
              hor_sync<='0';
              cnt:=cnt+1;
      elsif cnt<=799 then
              hor_sync<='1';
              cnt:=cnt+1;
      else
              cnt:=0;
      end if;
end if;
End Process;
END ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ใช้ในการกำเนิดสัญญาณ VERTICAL_SYNC

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
ENTITY ver_sync IS
        PORT (hcounter          : IN  bit;
                reset           : IN  bit;
                ver_sync        : OUT bit
                );
END ;
ARCHITECTURE ver_sync_a OF ver_sync IS
BEGIN
Process(hcounter)
Variable cnt : integer range 0 to 1023;
Begin
if hcounter'event and hcounter='1' then
 if reset = '1' then
          cnt:=0;
        elsif cnt <= 830 then
                ver_sync<='0';
                cnt:=cnt+1;
        elsif cnt<=832 then
                        ver_sync<='1';
                        cnt:=cnt+1;
        elsif cnt<=888 then
                        ver_sync<='0';
                        cnt:=cnt+1;
        else
                        cnt:=0;
        end if;
                        cnt:=cnt+1;
end if;
End Process;
END ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ใช้ในการกำเนิดสัญญาณสี RGB

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
ENTITY rgb IS
        PORT (clk       : IN bit;
              sw_in     : IN bit_vector(0 to 2);
              reset     : OUT bit;
              r,g,b     : OUT integer range 0 to 255
              );
END;
ARCHITECTURE rgb_a OF rgb IS
BEGIN
Process(clk)
Variable count:integer range 0 to 1023;
Variable cr,cb:integer range 0 to 255;
Variable cnt:integer range 0 to 3;
variable flag0,flag1,flag2,flag3:integer range 0 to 3;
Begin
if clk'event and clk='1' then
--***test_w***--
 if sw_in = "001" then
        if flag0 < 3 then
              flag0 := flag0+1;
              flag1 := 0;       flag2 := 0;       flag3 := 0;
              reset <= '1';
              count:=0;
              cnt:=0;           cb:=0;
        elsif count<640 then
              if cnt=3 then
                    r<=cb;  g<=cb;
                    b<=cb;  cb:=cb+1;
              end if;
              count:=count+1;
              reset <= '0';
        elsif count<800 then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                r<=0;     g<=0;
                b<=0;     cb:=0;
                count:=count+1;
                reset <= '0';
            else
                count:=0;
                reset <= '0';
            end if;
            cnt:=cnt+1;
--***test_r***--
  elsif sw_in = "010" then
            if flag1 < 3 then
                flag1 := flag1+1;
                flag0 := 0;          flag2 := 0;          flag3 := 0;
                reset <= '1';
                count:=0;
                cnt:=0;              cb:=0;
            elsif count<640 then
                if cnt=3 then
                    r<=cb;    g<=0;
                    b<=0;     cb:=cb+1;
                end if;
                count:=count+1;
                reset <= '0';
            elsif count<800 then
                r<=0;     g<=0;
                b<=0;     cb:=0;
                count:=count+1;
                reset <= '0';
            else
                count:=0;
                reset <= '0';
            end if;
            cnt:=cnt+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
--***test_g***--
 elsif sw_in = "011" then
                if flag2 < 3 then
                    flag2 := flag2+1;
                    flag0 := 0;          flag1 := 0;          flag3 := 0;
                    reset <= '1';
                    count:=0;
                    cnt:=0;              cb:=0;
                elsif count<640 then
                    if cnt=3 then
                            r<=0;    g<=cb;
                            b<=0;    cb:=cb+1;
                    end if;
                    count:=count+1;
                    reset <= '0';
                elsif count<800 then
                    r<=0;    g<=0;
                    b<=0;    cb:=0;
                    count:=count+1;
                    reset <= '0';
                else
                    count:=0;
                    reset <= '0';
                end if;
                cnt:=cnt+1;
--***test_b***--
 elsif sw_in = "100" then
                if flag3 < 3 then
                    flag3 := flag3+1;
                    flag0 := 0;          flag1 := 0;          flag2 := 0;
                    reset <= '1';
                    count:=0;
                    cnt:=0;              cb:=0;
                elsif count<640 then
                    if cnt=3 then
                            r<=0;    g<=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    b<=cb;    cb:=cb+1;
                end if;
                count:=count+1;
                reset <= '0';
            elsif count<800 then
                r<=0;     g<=0;
                b<=0;     cb:=0;
                count:=count+1;
                reset <= '0';
            else
                count:=0;
                reset <= '0';
            end if;
            cnt:=cnt+1;
--***test_rg***--
  elsif sw_in = "101" then
            if count<640 then
                if cnt=3 then
                    r<=cr;     g<=0;
                    b<=cb;    cb:=cb+1;
                    cr:=cr-1;
                end if;
                count:=count+1;
                reset <= '0';
            elsif count<800 then
                r<=0;     g<=0;
                b<=0;
                cb:=0;    cr:=255;
                count:=count+1;
                reset <= '0';
            else
                count:=0;
                reset <= '0';
            end if;
            cnt:=cnt+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
--***test_gb***--
  elsif sw_in = "110" then
                if count<640 then
                    if cnt=3 then
                            r<=cr;    g<=cb;
                            b<=0;     cb:=cb+1;
                            cr:=cr-1;
                    end if;
                    count:=count+1;
                    reset <= '0';
                elsif count<800 then
                    r<=0;     g<=0;
                    b<=0;
                    cb:=0;    cr:=255;
                    count:=count+1;
                    reset <= '0';
                else
                    count:=0;
                    reset <= '0';
                end if;
                cnt:=cnt+1;
--***test_br***--
elsif sw_in = "111" then
                if count<640 then
                    if cnt=3 then
                            r<=0;     g<=cr;
                            b<=cb;    cb:=cb+1;
                            cr:=cr-1;
                    end if;
                    count:=count+1;
                    reset <= '0';
                elsif count<800 then
                    r<=0;     g<=0;
                    b<=0;
                    cb:=0;    cr:=255;
                    count:=count+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                reset <= '0';
            else
                count:=0;
                reset <= '0';
            end if;
            cnt:=cnt+1;
  end if;
end if;
End Process;
END ;
```

โปรแกรมที่ใช้ในการเลือกสวิทช์

```
Library ieee;
Use ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
Entity selects is
Port        (   sw      :       in bit_vector(0 to 6);
                output  :       out     bit_vector(0 to 2)
            );
End ;
Architecture a_select of selects is
Begin
Process
begin
if sw(0) = '0' then
            output<="001";
elsif sw(1) = '0' then
            output<="010";
elsif sw(2) = '0' then
            output<="011";
elsif sw(3) = '0' then
            output<="100";
elsif sw(4) = '0' then
            output<="101";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
elsif sw(5) = '0' then
                output<="110";
elsif sw(6) = '0' then
                output<="111";
end if;
end process;
end;
```

โปรแกรมที่ใช้ในการเลือกขับ LED

```
Library ieee;
Use ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
Entity led is
Port        (    sw_led   :        in bit_vector(0 to 2);
                 led1,led2,led3,led4,led5,led6,led7    :        out      bit
                 );
End ;
Architecture a_led of led is
Begin
Process
begin
if sw_led = "001" then
                led1<='0'; led2<='1'; led3<='1'; led4<='1';
                led5<='1'; led6<='1'; led7<='1';
elsif sw_led = "010" then
                led1<='1'; led2<='0'; led3<='1'; led4<='1';
                led5<='1'; led6<='1'; led7<='1';
elsif sw_led = "011" then
                led1<='1'; led2<='1'; led3<='0'; led4<='1';
                led5<='1'; led6<='1'; led7<='1';
elsif sw_led = "100" then
                led1<='1'; led2<='1'; led3<='1'; led4<='0';
                led5<='1'; led6<='1'; led7<='1';
elsif sw_led = "101" then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                led1<='1'; led2<='1'; led3<='1'; led4<='1';
                led5<='0'; led6<='1'; led7<='1';
elsif sw_led = "110" then
                led1<='1'; led2<='1'; led3<='1'; led4<='1';
                led5<='1'; led6<='0'; led7<='1';
elsif sw_led = "111" then
                led1<='1'; led2<='1'; led3<='1'; led4<='1';
                led5<='1'; led6<='1'; led7<='0';
end if;
end process;
end;
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

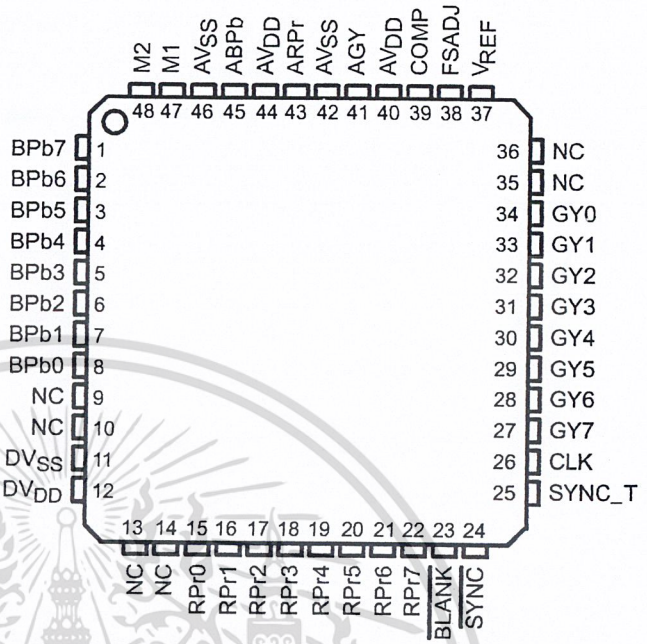
features

- Triple 8-Bit D/A Converters
- Minimum 80 MSPS Operation
- Direct Drive of Doubly-Terminated 75-Ω Load Into Standard Video Levels
- 3×8 Bit 4:4:4, 2×8 Bit 4:2:2 or 1×8 Bit 4:2:2 (ITU-BT.656) Multiplexed YPbPr/GBR Input Modes
- Bi-Level (EIA) or Tri-Level (SMPTE) Sync Generation With 7:3 Video/Sync Ratio
- Integrated Insertion of Sync-On-Green/ Luminance or Sync-On-All Channels
- Configurable Blanking Level
- Internal Voltage Reference

applications

- High-Definition Television (HDTV) Set-Top Boxes/Receivers
- High-Resolution Image Processing
- Desktop Publishing
- Direct Digital Synthesis/I-Q Modulation

TQFP-48 PowerPAD™ PACKAGE
(TOP VIEW)



See ALSO: THS8133 (10 bit, pin-compatible)

description

The THS8134 is a general-purpose triple high-speed D/A converter (DAC) optimized for use in video/graphics applications. The device operates from a 5-V analog supply and a 3-V to 5-V range digital supply. The THS8134 has a sampling rate up to 80 MSPS. The device consists of three 8-bit D/A converters and additional circuitry for bi-level/tri-level sync and blanking level generation in video applications.

THS8134 is also well-suited in applications where multiple well-matched and synchronously operating DACs are needed; for example, I-Q modulation and direct-digital synthesis in communications equipment.

The current-steering DACs can be directly terminated in resistive loads to produce voltage outputs. The device provides a flexible configuration of maximum output current drive. Its output drivers are specifically designed to produce standard video output levels when directly connected to a single-ended doubly-terminated 75 Ω coaxial cable. Full-scale video/sync is generated in a 7:3 ratio, compliant with SMPTE standards for GBR and YPbPr signals.

Furthermore, the THS8134 can generate both a traditional bi-level sync or a tri-level sync signal, as per the SMPTE standards, via a digital control interface. The sync signal is inserted on one of the analog output channels (sync-on-green/luminance) or on all output channels. Also, a blanking control signal sets the outputs to defined levels during the nonactive video window.

The position of this defined (blanking) level and the temperature range, over which the maximum imbalance between the inserted analog syncs ($K_{IMBAL(SYNC)}$), are the only differences between the unrev, revA, and revB device versions. Refer to the Available Options table.

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.

PowerPAD is a trademark of Texas Instruments.

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 © 2000, Texas Instruments Incorporated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

escription (continued)

Finally the input format can be either 3×8 bit 4:4:4, 2×8 bit 4:2:2, or 1×8 bit 4:2:2. This enables a direct interface to a wide range of video DSP/ASICs including parts generating ITU-BT.656 formatted output data.

AVAILABLE OPTIONS

T_A	PACKAGE
	TQFP-48 PowerPAD
0°C to 70°C	THS8134CPHP†
	THS8134ACPHP‡
	THS8134BCPHP‡

† In the THS8134CPHP, the K_{IMBAL} maximum specification is assured over full temperature range and the $K_{IMBAL(SYNC)}$ maximum specification is assured at 25°C. The position of the blanking level is as shown in Table 1.

‡ In the THS8134ACPHP and the THS8134BCPHP, both the K_{IMBAL} maximum specification and the $K_{IMBAL(SYNC)}$ maximum specification are assured over the full temperature range. The position of the blanking level is as shown in Table 1.

Terminal Functions

TERMINAL NAME	PIN	I/O	DESCRIPTION
3Pb	45	O	Analog red, green and blue respectively Pr, Y and Pb current outputs, capable of directly driving a doubly terminated 75-Ω coaxial cable.
GY	41	O	
RPr	43	O	
/DD	40,44	I	Analog power supply (5 V ±10%). All AV_{DD} terminals must be connected.
/SS	42,46	I	Analog ground
LANK	23	I	Blanking control input, active low. A rising edge on CLK latches $\overline{BLANK}$. When asserted, the ARPr, AGY and ABPb outputs are driven to the blanking level, irrespective of the value on the data inputs. $\overline{SYNC}$ takes precedence over $\overline{BLANK}$, so asserting $\overline{SYNC}$ (low) while $\overline{BLANK}$ is active (low) will result in sync generation.
Pb0–BPb7	8–1	I	Blue or Pb pixel data input bus. Index 0 denotes the least significant bit. Refer to functional description for different operating modes.
LK	26	I	Clock input. A rising edge on CLK latches RPr0-7, GY0-7, BPb0-7, $\overline{BLANK}$, $\overline{SYNC}$, and SYNC_T. The M2 input is latched by a rising edge on CLK also, but only when additional conditions are satisfied, as explained in its terminal description.
OMP	39	O	Compensation terminal. A 0.1 µF capacitor must be connected between COMP and AV_{DD}.
/DD	12	I	Digital power supply (3-V to 5-V range)
/SS	11	I	Digital ground
SADJ	38	I	Full-scale adjust control. The full-scale current drive on each of the output channels is determined by the value of a resistor R_{FS} connected between this terminal and AV_{SS}. The nominal value of R_{FS} is 430 Ω, corresponding to 26.67 mA full-scale current. The relationship between R_{FS} and the full-scale current level for each operation mode is explained in the functional description.
Y0–GY7	34–27	I	Green or Y pixel data input bus. Index 0 denotes the least significant bit. Refer to functional description for different operating modes.
1	47	I	Operation mode control 1. M1 is directly interpreted by the device (it is not latched by CLK). M1 configures device according to Table 1.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Terminal Functions (Continued)

TERMINAL NAME	TERMINAL PIN	I/O	DESCRIPTION
M2	48	I	Operation mode control 2. The second rising edge on CLK after a transition on $\overline{\text{SYNC}}$ latches M2. The interpretation is dependent on the polarity of the last $\overline{\text{SYNC}}$ transition: $\overline{\text{SYNC}}$ L to H: latched as M2_INT $\overline{\text{SYNC}}$ H to L: latched as INS3_INT Together with M1, M2_INT configures the device as shown in Table 1. When INS3_INT is high, the sync output is inserted on all DAC outputs; a low will insert it only on the AGY output. See also Figure 2 and Table 2. The value of M2 at power-up is undetermined. Therefore at least 1 L → H transition on $\overline{\text{SYNC}}$ is required to set M2.
NC	9, 10, 13, 14, 35, 36		Not connected
RPr0–RPr7	15–22	I	Red or Pr pixel data input bus. Index 0 denotes the least significant bit. Refer to functional description for different operating modes.
$\overline{\text{SYNC}}$	24	I	Sync control input, active low. A rising edge on CLK latches $\overline{\text{SYNC}}$. When asserted, only the AGY output (INS3_INT=L, see terminal M2) or ARPr, AGY and ABPb outputs (INS3_INT=H, see terminal M2) are driven to the sync level, irrespective of the values on the data or $\overline{\text{BLANK}}$ inputs. Consequently, $\overline{\text{SYNC}}$ should remain low for the whole duration of sync, which is in the case of a tri-level sync both the negative and positive portion (see Figure 7).
SYNC_T	25	I	Sync tri-level control, active high. A rising edge on CLK latches SYNC_T. When asserted, a positive sync (higher than blanking level) is generated when $\overline{\text{SYNC}}$ is low. When disabled, a negative sync (lower than blanking level) is generated when $\overline{\text{SYNC}}$ is low. When generating a tri-level (negative-to-positive) sync, a L → H transition on this signal positions the start of the positive transition. See Figure 6 for timing control. The value on SYNC_T is ignored when $\overline{\text{SYNC}}$ is not asserted (high).
V_{REF}	37	I/O	Voltage reference for DACs. An internal voltage reference of nominally 1.35 V is provided, which requires an external 0.1 µF ceramic capacitor between V_{REF} and AV_{SS}. However, the internal reference can be overdriven by an externally supplied reference voltage.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

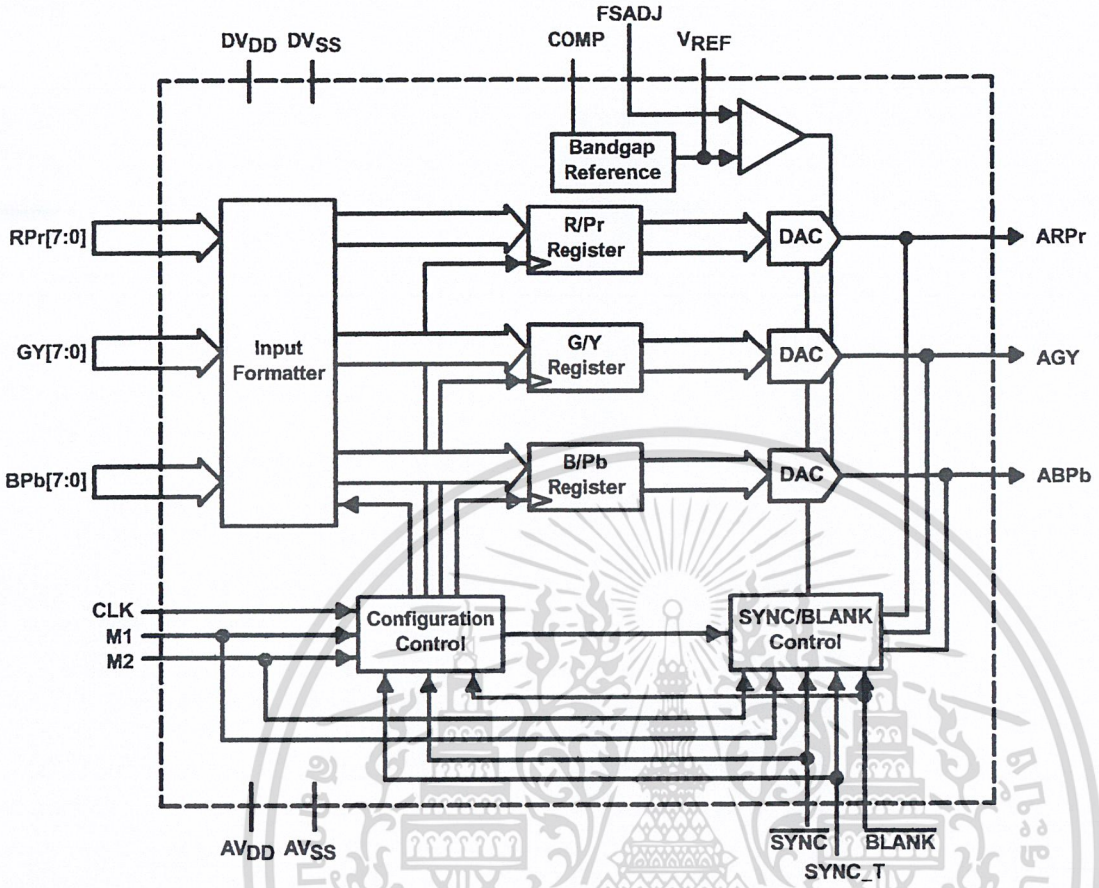


Figure 1. THS8134 Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำ[illegible]ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

functional description

device configuration

Input data to the device can be supplied from a 3x8b GBR/YPbPr input port. If the device is configured to take data from all three channels, the data is clocked in at each rising edge of CLK. All three DACs operate at the full clock speed of CLK.

In the case of 4:2:2 sampled data (for YPbPr) the device can be fed over either a 2x8 bit or 1x8 bit multiplexed input port. An internal demultiplexer will route input samples to the appropriate DAC: Y at the rate of CLK, Pb and Pr each at the rate of one-half CLK.

According to ITU-BT.656, the sample sequence is Pb-Y-Pr over a 1x8 bit interface (Y-port). The sample sequence starts at the first rising edge of CLK after $\overline{\text{BLANK}}$ has been taken high (inactive). In this case the frequency of CLK is two times the Y conversion speed and four times the conversion speed of both Pr and Pb.

With a 2x8 bit input interface, both the Y-port and the Pr-port are sampled on every CLK rising edge. The Pr-port carries the sample sequence Pb-Pr. The sample sequence starts at the first rising edge of CLK after $\overline{\text{BLANK}}$ has been taken high (inactive). In this case the frequency of CLK is equal to the conversion speed of Y and 2x the conversion speed of both Pr and Pb.

The device's operation mode is set by the M1 and M2 mode selection terminals, according to Table 1. The operation mode also determines the blanking level, as explained below in the sync/blanking generation sections.

Table 1. THS8134 Configuration

M1	M2_INT	CONFIGURATION	DESCRIPTION
L	L	GBR 3x8b–4:4:4	GBR mode 4:4:4. Data clocked in on each rising edge of CLK from G, B, and R input channels. For the definition of the analog output levels during blanking, see note 1.
L	H	YPbPr 3x8b–4:4:4	YPbPr mode 4:4:4. Data clocked in on each rising edge of CLK from Y, Pb and Pr input channels. For the definition of the analog output levels during blanking, see note 1.
H	L	YPbPr 2x8b–4:2:2	YPbPr mode 4:2:2 2x8 bit. Data clocked in on each rising edge of CLK from Y & Pr input channels. A sample sequence of Pb–Pr–... should be applied to the Pr port. At the first rising edge of CLK after $\overline{\text{BLANK}}$ is taken high, Pb should be present on this port. For the definition of the analog output levels during blanking, see note 1.
H	H	YPbPr 1x8b–4:2:2	YPbPr mode 4:2:2 1x8 bit (ITU-BT.656 compliant). Data clocked in on each rising edge of CLK from Y input channel. For the definition of the analog output levels during blanking, see note 1.

NOTE 1: In all device versions, the blanking level on the AGY channel output corresponds to input code 0 of the DAC.

- In the THS8134CPHP and the THS8134ACPHP versions, the blanking level on the ABPb and ARPr channel outputs corresponds to the 128 input code of the DAC, when sync is inserted on all three channels (INS3_INT=H), and to the 0 input code of the DAC, when sync is only inserted on the Y channel (INS3_INT=L).
- In the THS8134BCPHP version, the blanking level on the ABPb and ARPr channel outputs corresponds to the 128 input code of the DAC irrespective if sync is inserted on all three channels (INS3_INT=H), or if sync is inserted only on the Y channel (INS3_INT=L).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ[illegible]เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 2. INS3_INT/M2_INT Selection on M2

LAST VENT ON $\overline{\text{SYNC}}$	SYNC_T	M1	M2 (see Note 2)	DESCRIPTION
H→L	L or H	X	INS3_INT	Sync insertion active: $\overline{\text{SYNC}}$ low enables sync generation on 1 (INS3_INT=L) or all 3 (INS3_INT=H) DAC outputs. SYNC_T determines the sync polarity.
L→H	X	X	M2_INT	Device mode programming active: The DAC outputs reflect the DAC inputs ($\overline{\text{BLANK}}$=H) or are forced to the blanking level ($\overline{\text{BLANK}}$=L). M2 is interpreted according to Table 1.

Don't care

TE 2: M1 and M2 start configuring the device as soon as they are interpreted, which is continuously for M1 (static pin) or on the second rising edge on CLK after a transition on $\overline{\text{SYNC}}$ for M2. M2 is interpreted as either INS3_INT or M2_INT, as shown in Table 2.

ɔgramming example

Configuration of the device will normally be static in a given application. If M2_INT and INS3_INT need to be both low or high, the M2 pin is simply tied low or high. If M2_INT and INS3_INT need to have different levels, these can be easily derived from the signal on the $\overline{\text{SYNC}}$ pin, as shown in Table 3 and Figure 2.

Table 3. Generating M2 From $\overline{\text{SYNC}}$

In order to have:		Apply to M2:
M2_INT	INS3_INT	
L	H	...$\overline{\text{SYNC}}$ delayed by 2 CLK periods
H	L	...inverted $\overline{\text{SYNC}}$ delayed by 2 CLK periods

The input formats and latencies are shown in Figures 3–5 for each operation mode.

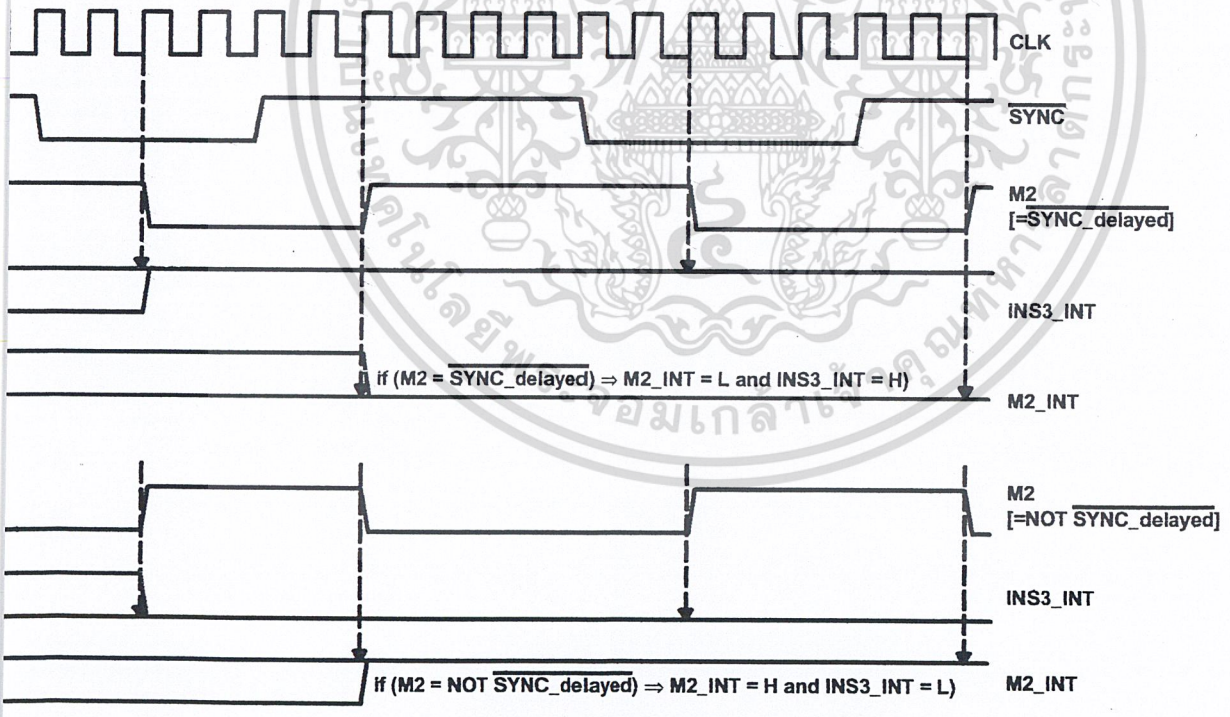


Figure 2. Generating INS3_INT and M2_INT from M2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

programming example (continued)

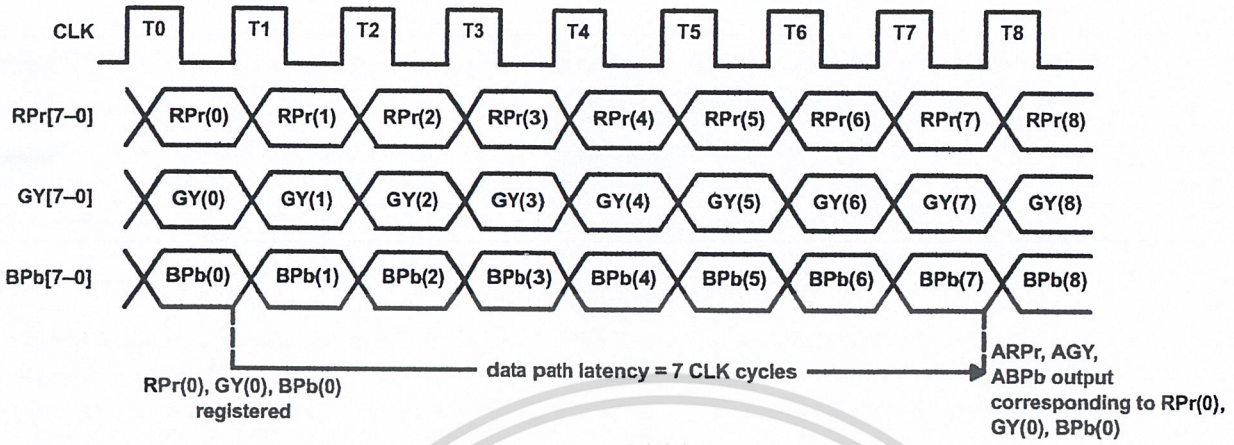


Figure 3. Input Format and Latency YPbPr 4:4:4 and GBR 4:4:4 Modes

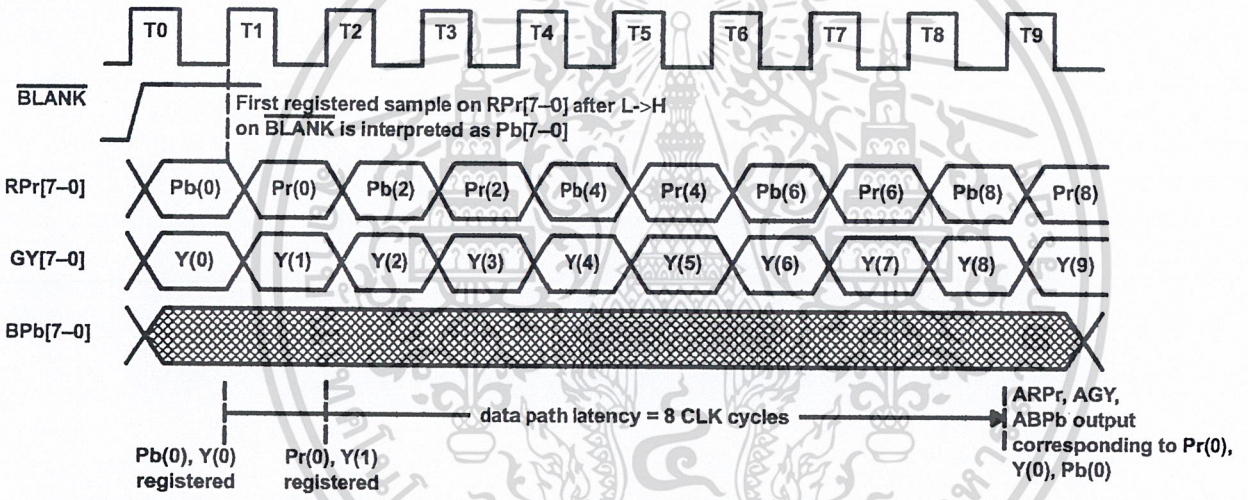


Figure 4. Input Format and Latency YPbPr 4:2:2 2×8 bit Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

rogramming example (continued)

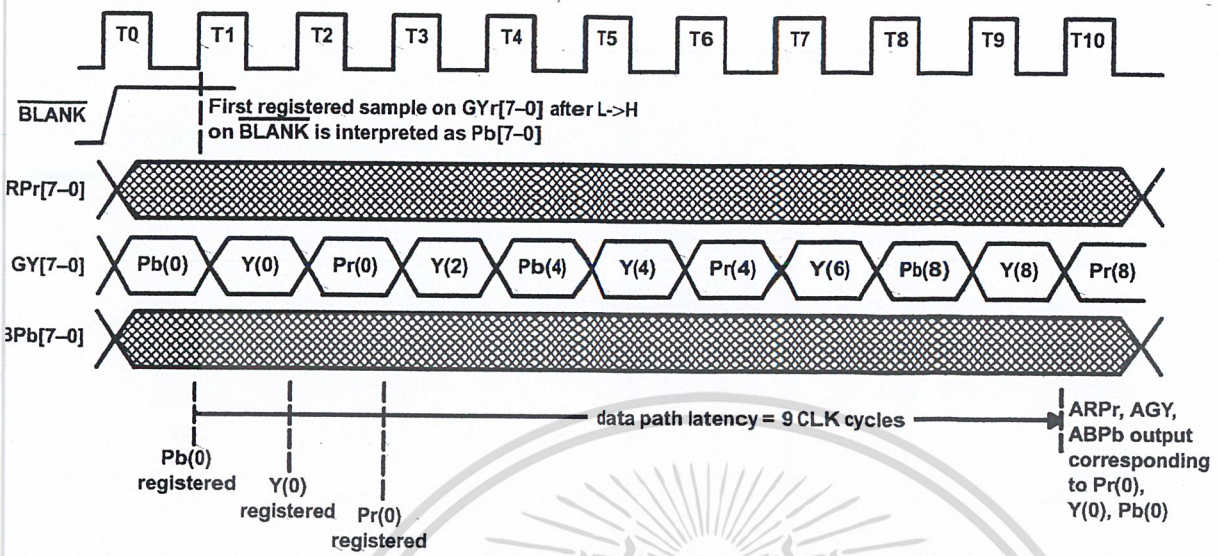


Figure 5. Input Format and Latency YPbPr 4:2:2 1×8 bit Mode

nc generation

Additional control inputs $\overline{\text{SYNC}}$ and SYNC_T enable the superposition of an additional current onto the AGY channel or on all three channels, depending on the setting of INS3_INT. By combining the $\overline{\text{SYNC}}$ and SYNC_T control inputs, either bi-level negative going pulses or tri-level pulses can be generated. Depending on the timing controls for these signals, both horizontal and vertical sync signals can be generated. Assertion of $\overline{\text{SYNC}}$ (active low) will identify the sync period, while assertion of SYNC_T (active high) within this period will identify the positive excursion of a tri-level sync.

Refer to the application information section for practical examples on the use of these control inputs for sync generation.

anking generation

An additional control input $\overline{\text{BLANK}}$ is provided that will fix the output amplitude on all channels to the blanking level, irrespective of the value on the data input ports. However, sync generation has precedence over blanking; that is, if $\overline{\text{SYNC}}$ is low, the level of $\overline{\text{BLANK}}$ is *don't care*. The absolute amplitude of the blanking level with respect to active video is determined by the GBR or YPbPr operation mode of the device. Refer to the application information section for practical examples on the use of this control input for blank generation.

Figure 6 shows how to control $\overline{\text{SYNC}}$, SYNC_T, and $\overline{\text{BLANK}}$ signals to generate tri-level sync levels and blanking at the DAC output. A bi-level (negative) sync is generated similarly by avoiding the positive transition on SYNC_T during $\overline{\text{SYNC}}$ low.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

blanking generation (continued)

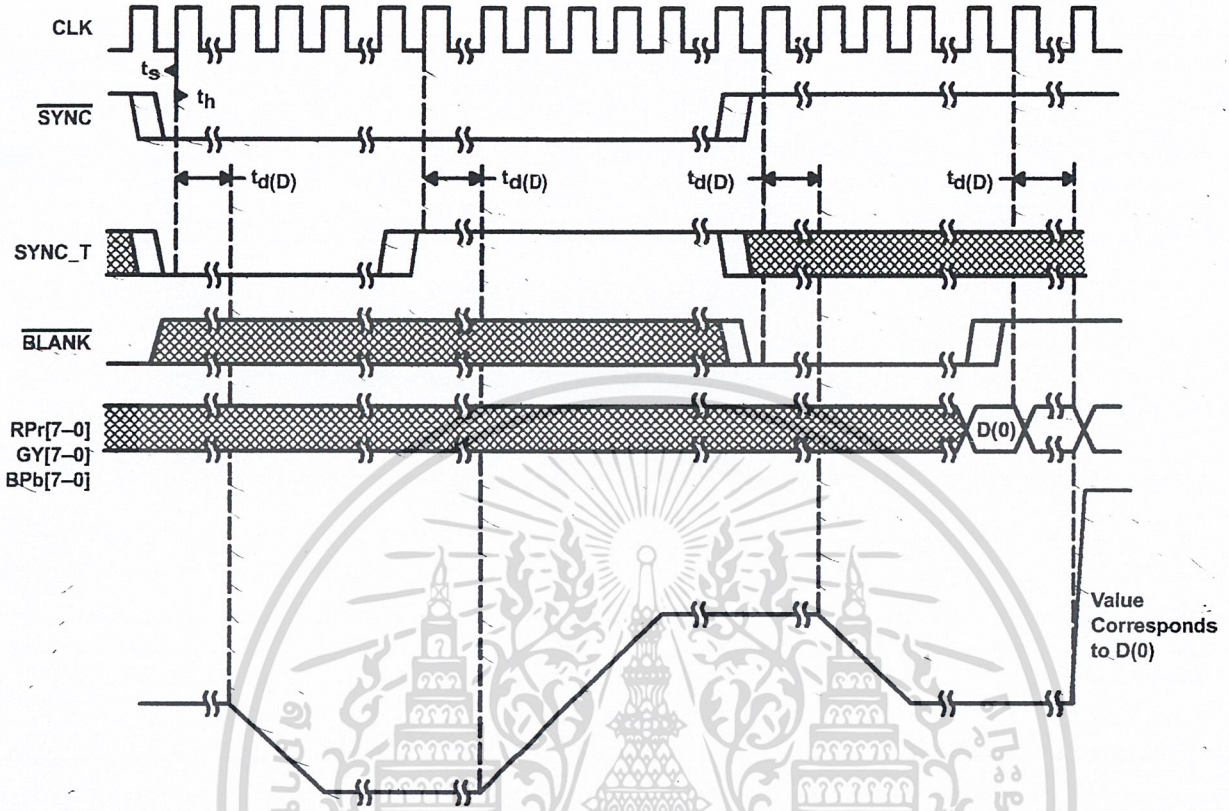


Figure 6. Sync and Blanking Generation

DAC operation

The analog output drivers generate a current of which the drive level can be user-modified by choice of an appropriate resistor value R_{FS}, connected to the FSADJ terminal. Refer to the paragraph on output amplitude control for details on how the output drive is affected by the operation mode of the device.

All current sources derive their amplitudes from an internal generator that produces a 1.35 V reference level. All current source amplitudes (video, blanking, sync) also come from this reference so that the relative amplitudes of sync/blank/video are always equal to their nominal relationships. For increased stability on the absolute levels, the user can overdrive the reference by directly driving the V_{REF} input terminal.

output amplitude control

The current drive on all three output channels and on the internal sync generator is controlled by a resistor R_{FS} that must be connected between FSADJ and AV_{SS}. In all operation modes the relative amplitudes of the current drivers are maintained irrespective of the R_{FS} value, as long as a maximum current drive capability is not exceeded.

The sync generator is composed of different current sources that are internally routed to a corresponding DAC output. Depending on the setting of INS3_INT during $\overline{\text{SYNC}}$ low, the sync current drive is added to either only the green channel output (sync-on-green) if INS3_INT = L or all three channel outputs INS3_INT = H. In either case the relative current levels, as defined below, are maintained.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ıtput amplitude control (continued)

The exact relationship between R_{FS} and the current drive level on each channel is dependent on the operation mode of the device (see Table 4). In GBR mode, the output drive is identical on the three channels, while in YPbPr mode, a level shift is implemented on Pb and Pr channels. Refer to the application information section for details on the current drive levels in each mode.

The device has an internal voltage reference derived from a bandgap reference of 1.35 V. The relationship between the full-scale current drive level and R_{FS} is given by:

$$I_{FS}\,[A] = \alpha \times V_{REF}\,[V] \div R_{FS}\,[\Omega]$$

where α is dependent on the operation mode of the device.

Typical operation modes are shown in Table 4 for the nominal R_{FS} value. This value will produce the full-scale current levels mentioned in Table 4 and, when terminated, voltages of standard video levels, as shown in the applications section. The resistor value is variable provided the maximum current level on each of the DAC outputs is not exceeded.

Table 4. THS8134 Nominal Full-Scale Currents

OPERATION MODE DESCRIPTION	M1	M2_INT	INS3_INT	AGY		ARPr		ABPb	
				I_{FS} (mA)	α	I_{FS} (mA)	α	I_{FS} (mA)	α
3R with sync-on-green	L	L	L	26.67†	1461/172	18.67‡	1023/172	18.67	1023/172
3R with sync-on-all	L	L	H	26.67	1461/172	26.67	1461/172	26.67	1461/172
PbPr with sync-on-Y	(L,H), (H,L) or (HH), according to Tables 1 and 2		L	26.67	1461/172	18.67	1023/172	18.67	1023/172
PbPr with sync-on-all			H	26.67	1461/172	18.67	1023/172	18.67	1023/172

S = 1461/172 × 1.35/430
S = 1023/172 × 1.35/430

solute maximum ratings over operating free-air temperature (unless otherwise noted)§

Supply voltage: AV_{DD} to AV_{SS}, DV_{DD} to DV_{SS} ..... –0.5 V to 7 V
AV_{DD} to DV_{DD}, AV_{SS} to DV_{SS} ..... –0.5 V to 0.5 V
Digital input voltage range to DV_{SS} ..... –0.5 V to DV_{DD} + 0.5 V
Operating free-air temperature range, T_A ..... 0°C to 70°C
Storage temperature range, T_{stg} ..... –55°C to 150°C

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

recommended operating conditions over operating free-air temperature range, T_A

power supply

		MIN	NOM	MAX	UNIT
Supply voltage	AV_{DD}	4.75	5	5.25	V
	DV_{DD}	3	3.3/5	5.25	

digital and reference inputs

		MIN	NOM	MAX	UNIT
High-level input voltage, V_{IH}	DV_{DD} = 3.3 V	2		DV_{DD}	V
	DV_{DD} = 5 V	2.4		DV_{DD}	
Low–level input voltage, V_{IL}		DV_{SS}		0.8	V
Clock frequency, f_{clk}		0		80	MHz
Pulse duration, clock high, $t_{w(CLKH)}$		5			ns
Pulse duration, clock low, $t_{w(CLKL)}$		5			ns
Reference input voltage,† $V_{ref(I)}$ (see Note 3)			1.35	1.62	V
FSADJ resistor, $R_{(FS)}$ (see Note 3)		360	430		Ω

† Voltage reference input applies to the externally applied voltage (overdrive condition). Internally a 2 kΩ resistor isolates the internal reference from the externally applied voltage, if any.

NOTE 3: The combination of V_{ref} and R_{FS} can be chosen at will as long as the maximum full-scale DAC output current $I_{(FS)}$ does not exceed 120% of its nominal value. Therefore, at fixed $R_{(FS)} = R_{(FSnom)}$, V_{ref} should not be higher than the maximum value mentioned and at fixed $V_{ref} = V_{ref(nom)}$, $R_{(FS)}$ should not be less than the minimum value mentioned.

electrical characteristics over recommended operating conditions with f_{CLK} = 80 MSPS and use of internal reference voltage V_{ref}, with $R_{(FS)} = R_{(FSnom)}$ (unless otherwise noted)

power supply (1 MHz, –1 dBFS digital sine simultaneously applied to all 3 channels)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
I_{DD}	Operating supply current	$AV_{DD} = DV_{DD}$ = 5 V		134	142	mA
		AV_{DD} = 5 V, DV_{DD} = 3.3 V		114	121	
P_D	Power dissipation	$AV_{DD} = DV_{DD}$ = 5 V		670	710	mW
		AV_{DD} = 5 V, DV_{DD} = 3.3 V		525	565	

power supply (1 MHz, –1 dBFS digital sine simultaneously applied to all 3 channels)

PARAMETER		UNIT	TEST CONDITIONS	MIN	TYP	MAX	UNIT
I_{DD}	Operating supply current	THS8134CPHP	$AV_{DD} = DV_{DD}$ = 5 V		134	142	mA
		THS8134ACPHP	AV_{DD} = 5 V, DV_{DD} = 3.3 V		114	121	
		THS8134BCPHP	$AV_{DD} = DV_{DD}$ = 5 V		159	167	mA
			AV_{DD} = 5 V, DV_{DD} = 3.3 V		136	143	mA
P_D	Power dissipation	THS8134CPHP	$AV_{DD} = DV_{DD}$ = 5 V		670	710	mW
		THS8134ACPHP	AV_{DD} = 5 V, DV_{DD} = 3.3 V		525	565	
		THS8134BCPHP	$AV_{DD} = DV_{DD}$ = 5 V		790	835	mA
			AV_{DD} = 5 V, DV_{DD} = 3.3 V		635	675	mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ[illegible]เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ectrical characteristics over recommended operating conditions with f_{CLK} = 80 MSPS and use f internal reference voltage V_{ref}, with $R_{(FS)} = R_{(FSnom)}$ (unless otherwise noted) (continued)

gital inputs – dc characteristics

	PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
H	High-level input current	$AV_{DD} = DV_{DD} = 5$ V Digital inputs and CLK at 0 V for I_{IL}; Digital inputs and CLK at 5 V for I_{IH}			1	µA
_	Low-level input current		–1			µA
_(CLK)	Low-level input current, CLK		–1		1	µA
H(CLK)	High-level input current, CLK					
I	Input capacitance	T_A = 25°C		7		pF
	Data and control inputs setup time		3			ns
	Data and control inputs hold time		0			ns
(D)	Digital process delay from first registered color component of pixel† (see Figures 3–5)	RGB and YPbPr 4:4:4		7		CLK periods
		YPbPr 4:2:2 2×8 bit		8		
		YPbPr 4:2:2 1×8 bit		9		

his parameter is assured by design and not production tested. The digital process delay is defined as the number of CLK cycles required for ıe first registered color component of a pixel, starting from the time of registering it on the input bus, to propagate through all processing and ppear at the DAC output drivers. The remaining delay through the IC is the analog delay $t_{d(A)}$ of the analog output drivers.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

electrical characteristics over recommended operating conditions with f_{CLK} = 80 MSPS and use of internal reference voltage V_{ref}, with $R_{(FS)} = R_{(FSnom)}$ (unless otherwise noted) (continued)

analog (DAC) outputs

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
	DAC resolution			8	8		bits
INL	Integral nonlinearity	Static, best fit			±0.2	±1.2	LSB
DNL	Differential nonlinearity	Static			±0.2	±1	LSB
PSRR	Power supply ripple rejection ratio of DAC output (full scale)	f = 100 kHz (see Note 4)		37			dB
		f = 1 MHz (see Note 4)		43			
XTALK	Crosstalk between channels	f up to 30 MHz, (see Note 5)				–55	dB
$V_{O(ref)}$	Voltage reference output			1.30	1.35	1.40	V
$r_{o(VREF)}$	V_{REF} output resistance			7K	11K	15K	W
$G_{(DAC)}$	DAC gain factor				See Table 4		
	Imbalance between DACs, (K_{IMBAL})	See Note 6				±5%	
	Imbalance between positive and negative sync, ($K_{IMBAL(SYNC)}$)	See Note 6				±2%	
$V_{O(DAC)}$	DAC output compliance voltage (sync+video)	R_L = 37.5 Ω, See Note 7			1	1.2	V
		R_L = 75 Ω, See Note 7			2	2.4	
$I_{(FS)}$	GBR sync-on-green and YPbPr sync-on-Y/sync-on-all	Internal reference	AGY	24	26.67	28	mA
			ABPb and ARPr	17.3	18.67	19.7	
		External reference	AGY	24.9	26.67	27.2	
			ABPb and ARPr	17.5	18.67	19.3	
	GBR sync-on-all	Internal reference	AGY	24	26.67	28	mA
			ABPb and ARPr	24	26.67	28	
		External reference	AGY	24.9	26.67	27.2	
			ABPb and ARPr	24.9	26.67	27.2	
r_o	DAC output resistance	See Note 10		57		92	kΩ
C_O	DAC output capacitance (pin capacitance)				8		pF
$t_{r(DAC)}$	DAC output current rise time	10% to 90% of full scale			2		ns
$t_{f(DAC)}$	DAC output current fall time	10% to 90% of full scale			2		ns
$t_{d(A)}$	Analog output delay	Measured from CLK=$V_{IH(min)}$ to 50% of full-scale transition, See Note 8				9	ns
t_S	Analog output settling time	Measured from 50% of full scale transition on output to output settling, within 2%, See Note 9			5	9	ns
SNR	Signal -to-noise ratio	1 MHz, –1 dBFS digital sine input, measured from 0 MHz to 8.8 MHz			53		dB
SFDR	Spurious-free dynamic range	1 MHz, –1 dBFS digital sine input, measured from 0 MHz to 8.8 MHz			62		dB
BW(1 dB)	Bandwidth	See Note 11			40		MHz

NOTES:
4. PSRR is measured with a 0.1 µF capacitor between the COMP and AVDD terminal; with a 0.1 µF capacitor connected between the V_{REF} terminal and AVSS. The ripple amplitude is within the range 100 mVp-p to 500 mVp-p with the DAC output set to full scale and a double-terminated 75 Ω (=37.5 Ω) load. PSRR is defined as 20 × log(ripple voltage at DAC output/ripple voltage at AVDD input). Limits from characterization only.
5. Crosstalk spec applies to each possible pair of the 3 DAC outputs. Limits from characterization only.
6. The imbalance between DACs applies to all possible pairs of the three DACs. K_{IMBAL} is assured over full temperature range. In parts labeled THS8134CPHP, $K_{IMBAL(SYNC)}$ is assured at 25°C. In parts labeled THS8134ACPHP, $K_{IMBAL(SYNC)}$ is assured over the full temperature range.
7. Nominal values at $R_{(FS)} = R_{(FSnom)}$: Maximum values at $R_{(FS)} = R_{(FSnom)} \div 1.2$. Maximum limits from characterization only.
8. This value excludes the digital process delay, $t_{d(D)}$. Limit from characterization only.
9. Maximum limit from characterization only
10. Limit from characterization only
11. This bandwidth relates to the output amplitude variation in excess of the droop from the sinx/x sampled system. Since the output is a sample-and-hold signal, a $\sin(\pi \times F_{in} \div F_{clk}) \div (\pi \times F_{in} \div F_{clk})$ roll-off is observed, which accounts e.g. at F_{in} = 40 MHz and F_{clk} = 80 MSPS for –3.92 dB signal drop (*sync droop*). The total DAC output variation (*device droop*) consists of this and an additional amount (*excess droop*) caused by the output impedance of the device, as shown in Table 5.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ[illegible]เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

erformance plots of AGY output channel at 80 MSPS and use of internal reference

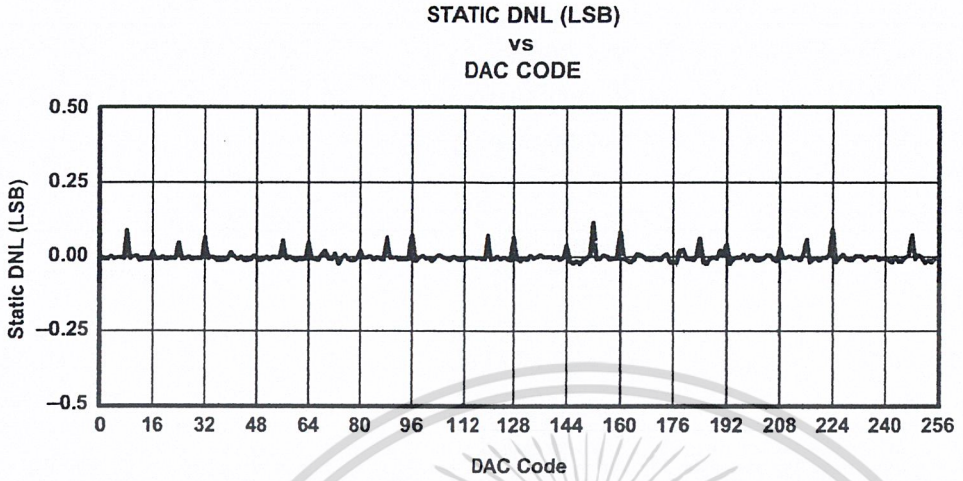


Figure 7. Static DNL

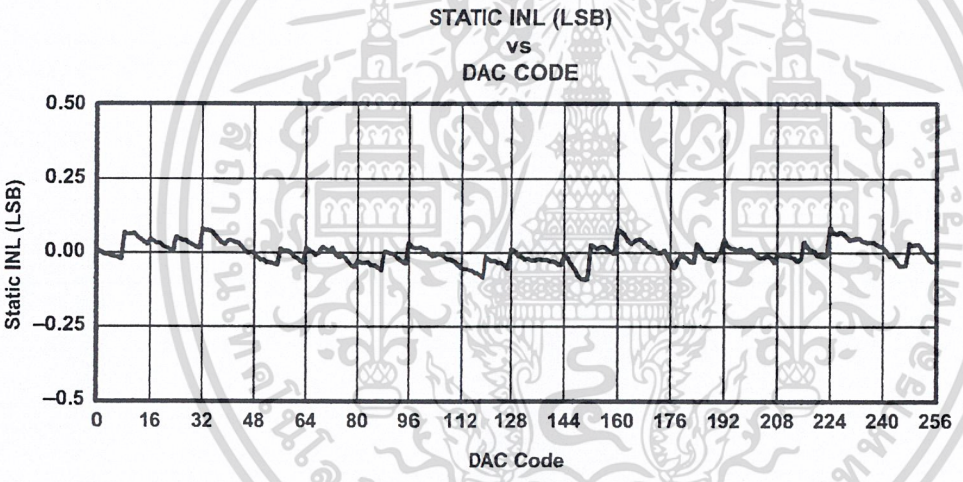


Figure 8. Static INL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

performance plots of AGY output channel at 80 MSPS and use of internal reference (continued)

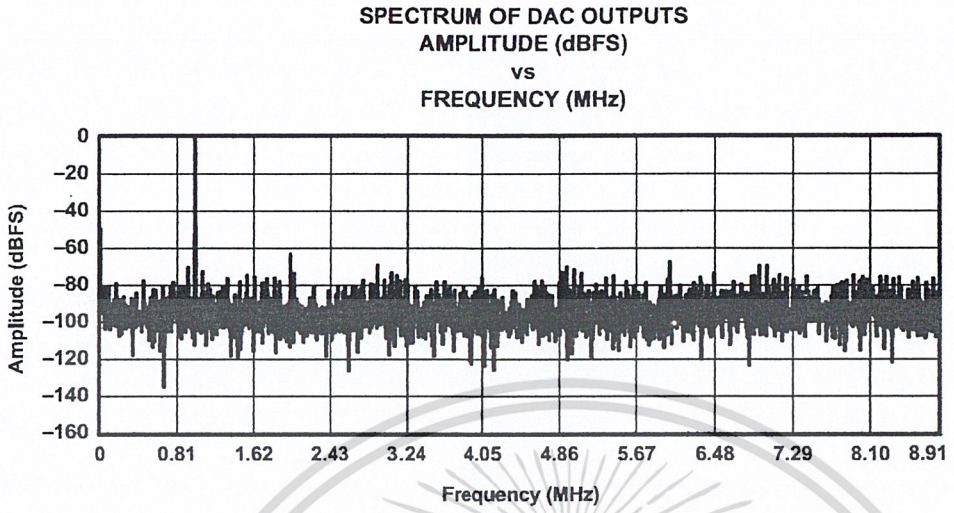


Figure 9. Spectral Plot for 1.02 MHz Digital Sine Input at 80 MSPS

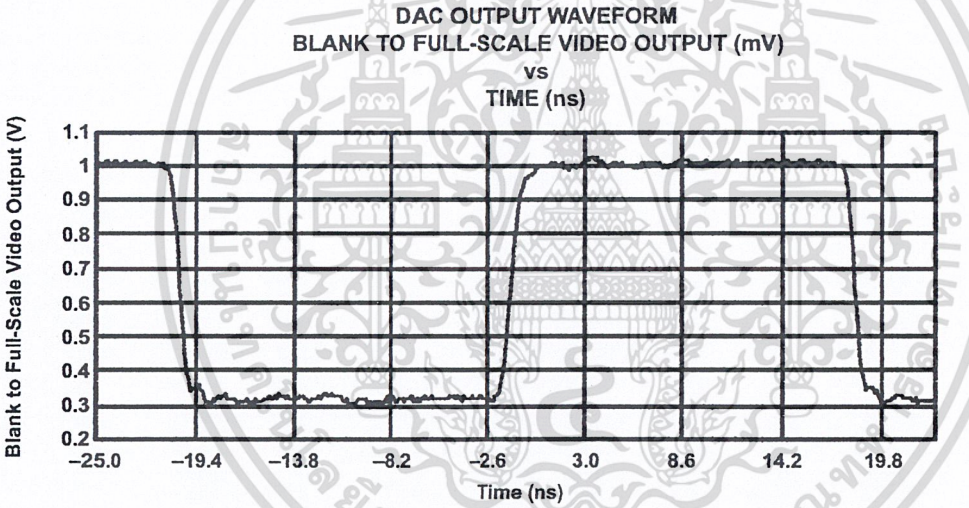


Figure 10. DAC Output Waveform (rise/fall and settling times)

Table 5. DAC Output Amplitude Variation Over Varying F_{in} at F_{clk} = 80 MSPS

F_{in} (kHz)	F_{clk} (MSPS)	SYNC DROOP (dB)	EXCESS DROOP (dB)
500	80	0	0
5000	80	–0.056	–0.02
10000	80	–0.22	–0.08
20000	80	–0.91	–0.29
30000	80	–2.11	–0.39
40000	80	–3.92	–0.40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION INFORMATION

ONFIGURING THS8134 FOR GENERATING SMPTE COMPLIANT SIGNALS

Table 6 lists the standards that relate to the definition of analog interfaces for component video signals.

Table 6. Relevant Video Standards

STANDARD	TITLE	SCOPE
MPTE 253M	3-channel RGB Analog Video Interface	Component analog video for studio applications using 525 lines, 59.94 fields, 2:1 interlace and 4:3 or 16:9 aspect ratio.
MPTE 274M	1920x1080 Scanning and Analog and Parallel Digital Interfaces for Multiple-Picture Rates	Definition of image format of 1920x1080 pixels inside a total raster of 1125 lines, with an aspect ratio of 16:9. Interlaced format used for 1080I display definition of the ATSC HDTV standard.
MPTE 296M	1280x720 Scanning, Analog and Digital Representation and Analog Interface	Definition of image format of 1280x720 pixels inside a total raster of 750 lines, with an aspect ratio of 16:9. Progressive format used for 720P display definition of the ATSC HDTV standard.

THS8134 can be used to generate output signals compliant to each of these standards. The configuration for each is detailed below. In each of the cases the current output of each DAC can be converted into standard-compliant voltage levels by connecting a double terminated 75Ω load, as shown in the top part of Figure 11.

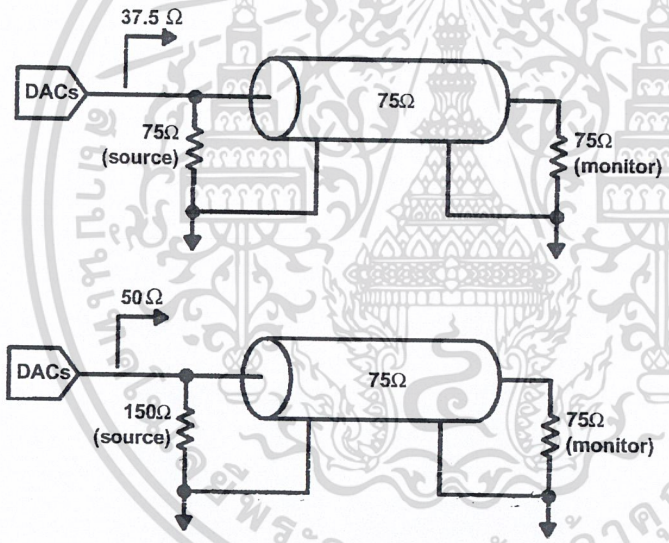


Figure 11. Typical Video Loads

The use of THS8134 for each of these standards is discussed next.

PTE 253M

This standard defines a component analog video interface using GBR color signals carried on parallel channels for the interconnection of television equipment. The scanning structure is typically 525 lines, 59.94 fields, 2:1 interlace and 4:3 or 16:9 aspect ratio. The analog signals of this standard are suitable for the generation of, or they can be generated from, digital video signals compliant to SMPTE 125M and SMPTE 267M by A/D or D/A conversion respectively. Furthermore SMPTE 253M signals can be the input to NTSC composite encoders compliant with SMPTE 170M. Table 7 lists the scope of the standards mentioned.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION INFORMATION

SMPTE 253M (continued)

Table 7. Video Standards Compatible with SMPTE 253M

STANDARD	TITLE
SMPTE 125M	Component Video Signal 4:2:2 – Bit-Parallel Digital Interface
SMPTE 267M	Bit – Parallel Digital Interface – Component Video Signal 4:2:2 16x9 Aspect Ratio
SMPTE 170M	Composite Analog Video Signal – NTSC for Studio Applications

The SMPTE 253M standard defines a GBR component set with positive going signals and a maximum peak level of 700 mV from blanking level. The green signal has a negative-going sync pulse of amplitude 300 mV from blanking level. The dc offset, as defined by the blanking level of the signal, is 0.0 V ±1.0 V. Figure 12 shows the waveform of the green channel, onto which the horizontal sync is inserted.

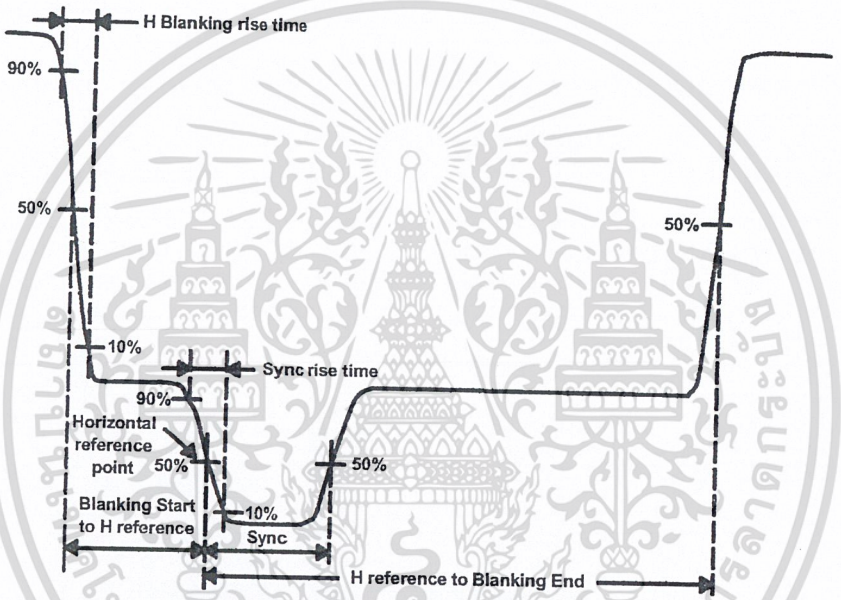


Figure 12. SMPTE 253M Line Waveform (green channel)

For this mode, the INS3_INT control should be kept low to enable sync-on-green only and the device is put in GBR 4:4:4 mode. This corresponds to the GBR with sync-on-green operation mode of Table 1.

Table 8 lists the THS8134 output currents that will produce compliant signals to this standard after proper termination, together with the required input signals.

Table 8. THS8134 Signals for SMPTE 253M Compliant Operation

LEVEL	AGY		ARPr, ABPb		SYNC	SYNC_T	BLANK	DAC INPUT
	(mA)	(V)	(mA)	(V)				
White	26.67	1.000	18.67	0.7000	1	X	1	FF_h
Video	Video+8.00	Video+0.3	Video	Video	1	X	1	data
Black	8.00	0.3000	0	0	1	X	1	00_h
Blank	8.00	0.3000	0	0	1	X	0	xx_h
Sync	0	0	0	0	0	0	x	xx_h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION INFORMATION

MPTE 253M (continued)

$\overline{\text{BLANK}}$ can be tied high in this mode if the data input is kept to 00h during the blanking time, since black and blanking level are at identical levels. Furthermore the SYNC_T terminal remains low, since only a bi-level sync is generated.

MPTE 274M

This standard defines a raster scanning format of 1920×1080 pixels inside a total raster of 1125 lines and an aspect ratio of 16:9, GBR and YPbPr color encoding formats and both analog and digital interfaces for GBR and YPbPr formats.

With respect to the analog interface, SMPTE 274M defines the position of the start of each line at the positive zero-crossing of a tri-level sync pulse. The sync pulse has a negative-going transition on a fixed number of clock cycles preceding this instant and another negative transition on a fixed number of clock cycles following this instant, as shown in Figure 13. The positive peak of sync is 300 mV; the negative peak of sync –300 mV.

The interface can carry both GBR o and YPbPr signals. The tri-level horizontal sync is inserted on all analog outputs and has identical absolute amplitude levels in all cases. For Y, black corresponds to a level of 0 V and peak white is 700 mV. Pb and Pr, on the other hand, have amplitudes between –350 mV and 350 mV.

The relative amplitudes of the current sources are identical to the case of SMPTE 253M. However, in this case a tri-level sync needs to be generated instead of a bi-level negative sync, and it needs to be present on all three component outputs. THS8134 supports the tri-level sync via an additional internal current source, activated by asserting SYNC_T. The sync insertion on all outputs is under the control of the INS3_INT pin. When asserted (high), the sync is inserted on all three output channels.

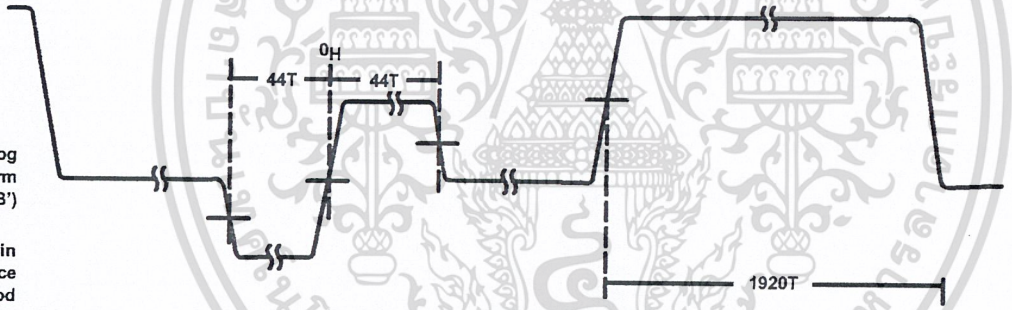


Figure 13. SMPTE 274M Line Waveform†

†This figure is for illustration purposes only. Consult the latest SMPTE 274M standard when designing a compliant system.

Figure 14 shows the relative amplitudes of video and horizontal/vertical sync. The level of vertical sync (broad pulse) is identical to the negative excursion of horizontal sync and therefore can be generated by the same current source on THS8134 by appropriately asserting the sync control inputs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION INFORMATION

SMPTE 274M (continued)

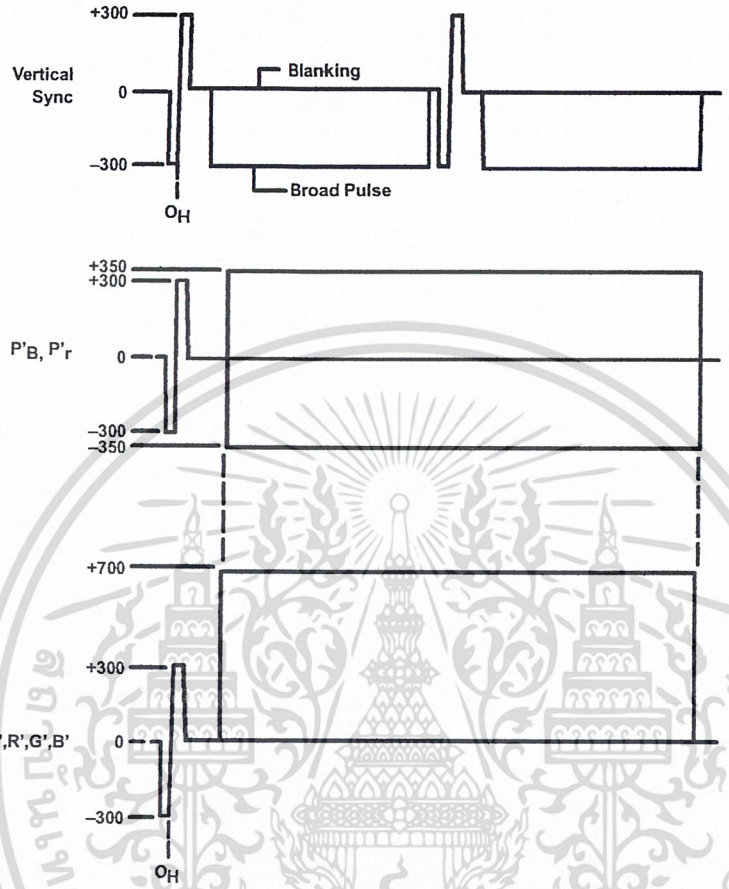


Figure 14. SMPTE 274M Analog Interface Horizontal Timing Details†

† This figure is for illustration purposes only. Consult the latest SMPTE 274M standard when designing a compliant system.

For GBR operation, Table 9 lists the THS8134 full-scale output currents that produce compliant signals to the standard after proper termination. These amplitudes are valid also in YPbPr mode for the Y channel. For GBR operation, the device needs to be configured with INS3_INT high, corresponding to the GBR with sync-on-all operation mode of Table 1.

Table 9. THS8134 Signals for SMPTE 274M Compliant Operation on GBR and Y Channels

LEVEL	GBRY		SYNC	SYNC_T	BLANK	DAC INPUT
	(mA)	(V)				
White	26.67	1.000	1	X	1	FF_h
Video	Video+8.00	Video+0.3	1	X	1	Data
Sync Pos	16.00	0.600	0	1	X	xx_h
Black	8.00	0.3000	1	X	1	00_h
Blank	8.00	0.3000	1	X	0	xx_h
Sync Neg	0	0	0	0	X	xx_h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION INFORMATION

MPTE 274M (continued)

In the YPbPr mode of this standard, the sync is centered around the center span of the video amplitude levels, as shown in Figure 14. So the current for Pb and Pr is down-shifted with respect to Y to accommodate the minimum data level at 0 mA. Thus, an input code of 00h corresponds now to an output drive of 0 mA while the negative sync level is at 1.33 mA, corresponding to 50 mV. The Pb and Pr data input format is offset binary.

Table 10 lists the THS8134 full-scale output currents for Pb and Pr channels in the YPbPr operation mode of the device. The operation mode corresponds to YPbPr with sync-on-all of Table 1.

Table 10. THS8134 Signals for SMPTE 274M Compliant Operation on Pb and Pr Channels

LEVEL	Pb, Pr		SYNC	SYNC_T	BLANK	DAC INPUT
	(mA)	(V)				
Max	18.67	0.7000	1	X	1	FF_h
Video	Video	Video	1	X	1	Data
Sync Pos	17.33	0.650	0	1	X	xx_h
Blank	9.33	0.350	1	X	0	xx_h
Sync Neg	1.33	0.050	0	0	x	xx_h
Min	0	0	1	X	1	00_h

MPTE 296M

This standard defines a raster scanning format of 1280x720 and an aspect ratio of 16:9, the analog and digital representation, and the definition of an analog interface. Both GBR and YPbPr component color encoding can be used.

With respect to the sync and video level definition, this standard is analogous to SMPTE 274M with the use of a tri-level sync pulse. Therefore, for the generation of output signals compliant to this standard, refer to the configuration of THS8134 for SMPTE 274M.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION INFORMATION

comparison to EIA RS-343/RS-170 levels

Traditionally, video amplitude levels are specified according to the EIA RS-343 or RS-170 standards. RS-343 uses a bi-level negative going sync. Also, there is a difference between the reference blanking and black video level. Figure 15 shows the relative amplitudes and the current drives that would be needed to generate compliant relative amplitudes with a double-terminated 75-Ω load, as is specified for RS-343. RS-170 compliant levels can be reached using the same current sources but a different 150-Ω source termination resistor, which brings the load to 150 || 75 Ω = 50 Ω. In this case a blank-to-white level of approximately 1 V is reached (0.714 V × 50 ÷ 37.5) as required by RS-170.

With Sync Insertion		Without Sync Insertion	
mA	V	mA	V
26.67	1.000	19.05	0.714
9.05	0.340	1.44	0.340
7.62	0.286	0	0
0	0		

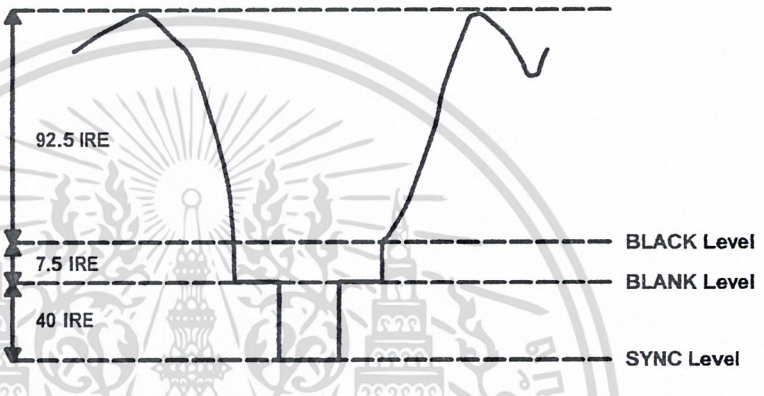


Figure 15. RS-343 Video Definition

The video signal contains 140 IRE, equal to 1 Vpp. This is split into 40 IRE for the composite sync, 7.5 IRE for blanking-to-black and 92.5 IRE for the active video portion.

designing with PowerPAD™

The THS8134 is housed in a high-performance, thermally enhanced, 48-pin PowerPAD package (TI package designator: 48-PHP). Use of the PowerPAD package does not require any special considerations except to note that the PowerPAD, which is an exposed die pad on the bottom of the device, is a metallic thermal and electrical conductor. Therefore, if not implementing the PowerPAD PCB features, solder masks (or other assembly techniques) may be required to prevent any inadvertent shorting by the exposed PowerPAD of connection etches or vias under the package. The recommended option, however, is not to run any etches or signal vias under the device, but to have only a grounded thermal land as explained below. Although the actual size of the exposed die pad may vary, the minimum size required for the keep-out area for the 48-pin PHP PowerPAD package is 7 mm × 7 mm.

It is recommended that there be a thermal land, which is an area of solder-tinned-copper, underneath the PowerPAD package. The thermal land will vary in size, depending on the PowerPAD package being used, the PCB construction, and the amount of heat that needs to be removed. In addition, the thermal land may or may not contain numerous thermal vias, depending on PCB construction.

More information on this package and other requirements for using thermal lands and thermal vias are detailed in the TI application note *PowerPAD™ Thermally Enhanced Package Application Report*, TI literature number SLMA002, available via the TI Web pages beginning at URL: http://www.ti.com.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION INFORMATION

esigning with PowerPAD™ (continued)

For the THS8134, this thermal land should be grounded to the low impedance ground plane of the device. This improves not only thermal performance but also the electrical grounding of the device. It is also recommended that the device ground terminal landing pads be connected directly to the grounded thermal land. The land size should be as large as possible without shorting device signal terminals. The thermal land may be soldered to the exposed PowerPAD using standard reflow soldering techniques.

While the thermal land may be electrically floated and configured to remove heat to an external heat sink, it is recommended that the thermal land be connected to the low impedance ground plane for the device.

Table 11 lists a comparison for thermal resistances between the PowerPAD package (48-PHP) used for this device and a regular 48-pin TQFP package (48-PFB).

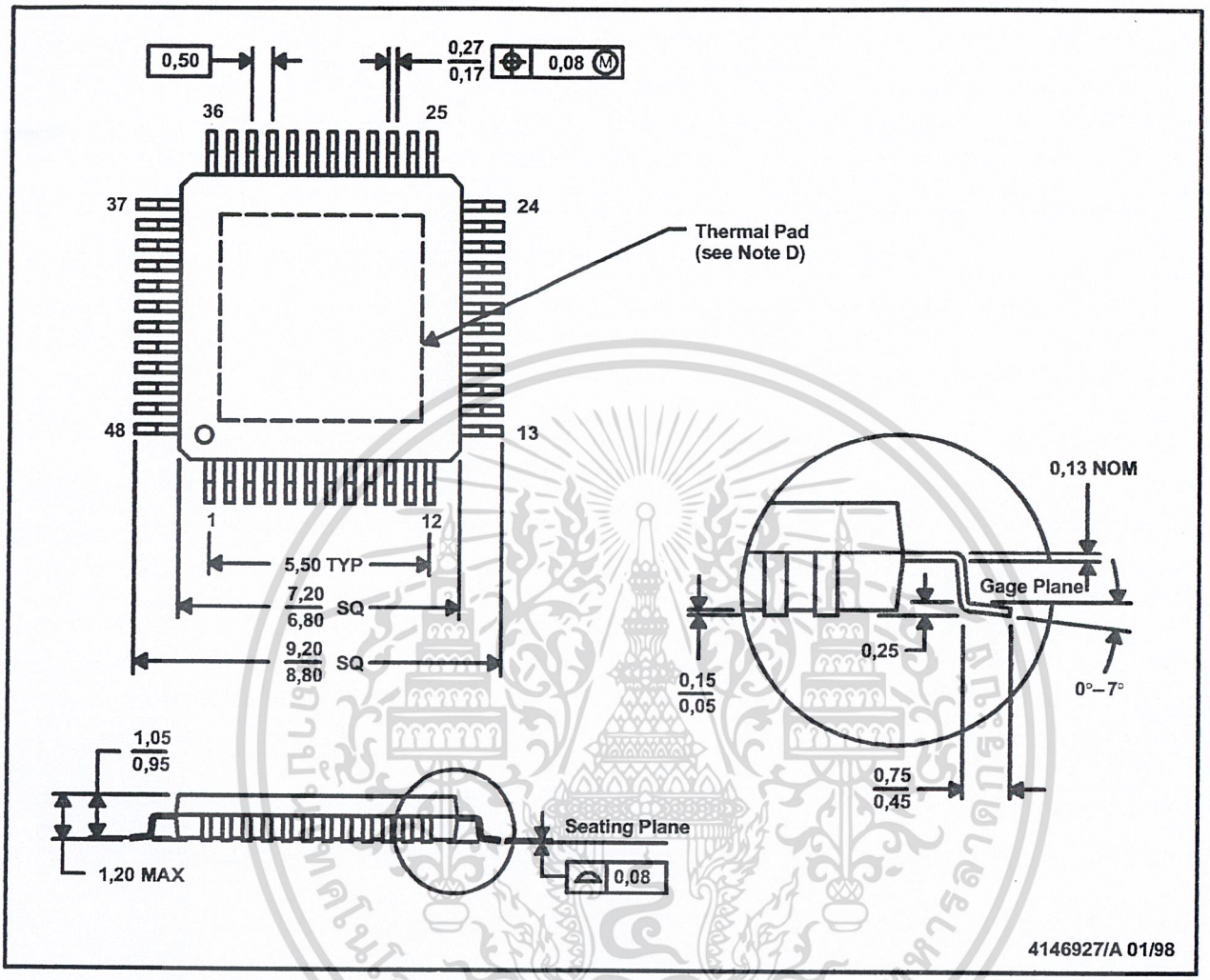
Table 11. Junction-Ambient and Junction-Case Thermal Resistances

48PHP PowerPAD vs 48PFB REGULAR TQFP	AIRFLOW IN lfm			
	0	150	250	500
θ_{JA} (°C/W) 48PHP	29.1	23.1	21.6	19.9
θ_{JC} (°C/W) 48PHP	1.14			
θ_{JA} (°C/W) 48PFB	97.5	78.3	71.6	63.5
θ_{JC} (°C/W) 48PFB	19.6			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MECHANICAL DATA

PHP (S-PQFP-G48) **PowerPAD™ PLASTIC QUAD FLATPACK**



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 protrusions.
D. The package thermal performance may be enhanced by bonding the thermal pad to an external thermal plane. This pad is electrically and thermally connected to the backside of the die and possibly selected leads.
E. Falls within JEDEC MO-153

PowerPAD is a trademark of Texas Instruments.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

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

Customers are responsible for their applications using TI components.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.

Copyright © 2000, Texas Instruments Incorporated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้