



รายงานสหกิจศึกษาฉบับสมบูรณ์

การลดเวลาการทดสอบแผงวงจรรวม
ด้วยวิธีการเพิ่มประสิทธิภาพหน่วยประมวลผลกลาง
Integrated Circuit Test Time Reduction
by New CPU Conversion

นางสาวอิสริย์ เซวงมหาปิติ

ภาควิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560



รายงานสหกิจศึกษาฉบับสมบูรณ์

การลดเวลาการทดสอบแผงวงจรรวม

ด้วยวิธีการเพิ่มประสิทธิภาพหน่วยประมวลผลกลาง

Integrated Circuit Test Time Reduction

by New CPU Conversion

นางสาวอิสริย์ เชนงมหาปิติ

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

หัวข้อโครงการสหกิจศึกษา	การลดเวลาการทดสอบแผงวงจรรวมด้วยวิธีการเพิ่มประสิทธิภาพ หน่วยประมวลผลกลาง
นักศึกษา	นางสาวอิสริย์ เขวงมหาปิติ รหัสประจำตัว 57011550
คณะ	วิศวกรรมศาสตร์ ภาควิชา อิเล็กทรอนิกส์
อาจารย์นิเทศ	อาจารย์ชินภัทร นันทจิวารักษ์ อาจารย์เกรียงไกร สุขสุด
ผู้นิเทศงาน	นายศรุต ศักดิ์บุรณพงษา
สถานประกอบการ	บริษัทผลิตอุปกรณ์เคมีคอนดักเตอร์

บทคัดย่อ

โครงการเล่มนี้อธิบายเกี่ยวกับการพัฒนาโปรแกรมที่ใช้ทดสอบคุณสมบัติของแผ่นเวเฟอร์ ให้ผลการทดสอบนั้นเป็นไปตามเกณฑ์มาตรฐานที่ต้องการ ซึ่งจะพัฒนาโปรแกรมจากที่ใช้หน่วยประมวลผลกลางแบบเก่ามาเป็นโปรแกรมที่ใช้หน่วยประมวลผลกลางแบบใหม่ โดยยังคงค่าประสิทธิภาพที่ดีในการทดสอบคุณสมบัติของแผ่นเวเฟอร์ไว้เท่ากับโปรแกรมดั้งเดิม การพัฒนาโปรแกรมดังกล่าวส่งผลให้ลดเวลาการทดสอบคุณสมบัติแผ่นเวเฟอร์ เนื่องจากหน่วยประมวลผลกลางแบบใหม่มีความเร็วในการประมวลผลสูงกว่าหน่วยประมวลผลกลางแบบเก่า นอกจากช่วยลดเวลาการทดสอบคุณสมบัติแผ่นเวเฟอร์แต่ละแผ่นให้เร็วขึ้นแล้ว ยังช่วยลดต้นทุนการทดสอบแผ่นเวเฟอร์ได้อีกด้วย ซึ่งรายละเอียดโดยรวมจะแบ่งออกเป็น 2 ส่วนใหญ่ๆ คือ ส่วนที่เป็นโปรแกรมกับส่วนของเครื่องทดสอบแผ่นเวเฟอร์

คำสำคัญ : โปรแกรมทดสอบคุณสมบัติของแผ่นเวเฟอร์, หน่วยประมวลผลกลางแบบเก่า, หน่วยประมวลผลกลางแบบใหม่

Cooperative Title Integrated Circuit Test Time Reduction by New CPU
Conversion

Student intern name Ms.Itsari Chawengmahapeti Student ID 57011550

Faculty Engineering **Department** Electronics

Advisor name Mr.Chinnapat Nantajiwakornchai
Mr.Kriangkrai Sooksood

Mentor name Mr.Sruit Sakburanapongsa

Company Semiconductor Manufacturing Company

ABSTRACT

This project is explaining about development of test program is used to test qualification of wafers and hope the results are in criteria. The new test program is developed from test program that used old CPU change into new CPU with the efficiency of qualification wafers results in new CPU test program are as same as qualification wafers results in old CPU test program. The new CPU test program development as above mentioned affects to reduce tested qualification of wafers time because new CPU has faster processing speed than old CPU. Not only, reduction of tested qualification of each wafers time, but also can save cost of wafer testing as well. Overall of details in this project can divide into two parts are the part of test program and wafer tester.

Keywords: Test program for testing qualification of wafers, Old CPU, New CPU

กิตติกรรมประกาศ

โครงการในครั้งนี้ สำเร็จลุล่วงไปด้วยความช่วยเหลืออย่างดียิ่งจาก นายศรุต ศักดิ์บุรณพงษ์ ซึ่งเป็นพนักงานที่ปรึกษาโครงการในครั้งนี้ โดยให้ความอนุเคราะห์ ให้คำแนะนำและคำปรึกษามาโดยตลอด ตั้งแต่ขั้นตอนแรกของการทำโครงการ เรียนรู้เกี่ยวกับโครงการที่ทำ จนถึงขั้นตอนที่ได้โครงการออกมาเสร็จสิ้นสมบูรณ์ ทำให้ข้าพเจ้าได้รับความรู้ ประสบการณ์ และความเข้าใจในการทำงานมากยิ่งขึ้น รวมถึงได้สอนให้รู้จักการทำงานกับผู้อื่น ในที่นี้คือได้ทำงานกับพี่ๆในProduction Line และสอนให้รู้จักแก้ไขปัญหาเฉพาะหน้า ต้องขอขอบคุณพี่ๆในฝ่ายที่ข้าพเจ้าได้ฝึกงาน ที่คอยช่วยเหลือ และดูแลข้าพเจ้าเป็นอย่างดีมาโดยตลอด คอยช่วยให้คำปรึกษาพร้อมแนะแนวทางแก้ปัญหา และข้าพเจ้าต้องขอขอบคุณทุกๆท่านที่ไม่ได้กล่าวไว้ในที่นี้อีกมากมาย ที่มีส่วนช่วยเหลือในการทำงานโครงการในครั้งนี้สำเร็จลุล่วงดังที่ได้ตั้งเป้าหมายไว้ทุกประการ

สารบัญ

ชื่อเรื่อง	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 สมมติฐานของการศึกษา.....	2
1.4 ขอบเขตของการวิจัย.....	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	4
2.1 เครื่องทดสอบ(tester).....	4
2.2 การเขียนโปรแกรมด้วยภาษา C++.....	9
บทที่ 3 วิธีการดำเนินงาน.....	12
3.1 ขั้นตอนการดำเนินโครงการ.....	12
บทที่ 4 ผลการทดสอบ.....	27
บทที่ 5 สรุปผลการทดสอบและข้อเสนอแนะ.....	28
เอกสารอ้างอิง	

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

ปัจจุบันนั้นโลกของเรามีการใช้เทคโนโลยีเข้ามาเพื่อสร้างความสะดวกสบายให้กับชีวิตประจำวันของเรามากขึ้น ซึ่งเทคโนโลยีเหล่านั้นส่วนใหญ่ได้นำมาใช้กับอุปกรณ์อิเล็กทรอนิกส์พวกเครื่องใช้ไฟฟ้าต่างๆ ส่งผลให้ตลาดธุรกิจทางด้านอิเล็กทรอนิกส์นั้นมีมากขึ้น เพื่อตอบสนองความต้องการของผู้คนที่ต้องการความสะดวกสบายจากการใช้อุปกรณ์เหล่านั้น ทำให้การแข่งขันของตลาดประเภทนี้ค่อนข้างสูง ดังนั้นบริษัทที่ทำเกี่ยวกับทางด้านนี้ควรหากลยุทธ์หรือวิธีการจัดการกับองค์กรของตนให้สามารถเติบโตได้และเป็นที่ต้องการของตลาดด้านอิเล็กทรอนิกส์ให้ได้ ดังเช่น บริษัทผลิตอุปกรณ์เซมิคอนดักเตอร์แห่งนี้ ซึ่งเป็นบริษัทที่ผลิตอุปกรณ์ทางเซมิคอนดักเตอร์เพื่อนำไปใช้กับเครื่องมือเครื่องใช้ต่างๆ ทางด้านอิเล็กทรอนิกส์ ได้เล็งเห็นถึงความสำคัญดังกล่าว จึงมีแนวคิดที่จะเพิ่มประสิทธิภาพการทำงานของเครื่องทดสอบงาน (tester) ให้ดีกว่าเดิม จึงได้เกิดเป็นโครงการ Integrated Circuit Test Time Reduction by New CPU Conversion นี้ขึ้นมา

Integrated Circuit Test Time Reduction by New CPU Conversion คือ การพัฒนาปรับปรุงโปรแกรมเก่าที่ใช้สำหรับทดสอบคุณสมบัติของแผ่นเวเฟอร์เพื่อใช้ในอุตสาหกรรมการผลิต โดยใช้หน่วยประมวลผลกลางแบบเก่า (Old CPU) มาเป็นโปรแกรมใหม่ที่ใช้หน่วยประมวลผลกลางแบบใหม่ (New CPU) โดยยังคงค่าประสิทธิภาพที่ดีในการทดสอบคุณสมบัติของแผ่นเวเฟอร์ไว้เท่ากับโปรแกรมดั้งเดิม ซึ่งประเด็นหลักที่ทำการพัฒนาโปรแกรมแบบนี้ขึ้นเพื่อช่วยลดเวลาการทดสอบคุณสมบัติของแผ่นเวเฟอร์แต่ละแผ่นและยังสามารถเพิ่มผลผลิตที่ได้จากการทดสอบงานได้อีกด้วย เนื่องจากหน่วยประมวลผลกลางแบบใหม่ (New CPU) นั้นมีความเร็วในการประมวลผลที่สูงกว่าหน่วยประมวลผลกลางแบบเก่า (Old CPU) โปรแกรมใหม่ที่สร้างขึ้นจากการเปลี่ยนหน่วยประมวลผลกลางมาเป็นแบบใหม่แล้วนั้น เรายังต้องทำการแก้ไขตัวโปรแกรมบางส่วนเพื่อให้เหมาะสมกับตัวหน่วยประมวลผลกลางแบบใหม่ด้วย ค่าที่อ่านได้ในตอนแรกจะไม่ถูกต้องตามที่ต้องการเนื่องจากการวัดในตำแหน่งที่ไม่เหมาะสม เกิดจากการประมวลผลที่เร็วเกินไปของหน่วยประมวลผลกลางแบบใหม่ (New CPU) ในการวัดค่าทางสถิตินั้นเราจะใช้เครื่องมือหนึ่งในการคำนวณค่าการวัดค่าพารามิเตอร์ต่างๆ ให้สามารถช่วยวิเคราะห์ข้อมูลได้ง่ายขึ้น เพื่อให้โปรแกรมใหม่ที่สร้างขึ้นคงค่าประสิทธิภาพไว้ได้ใกล้เคียงกับค่าเดิม

1.2 วัตถุประสงค์ของการวิจัย

1) เพื่อพัฒนาโปรแกรมที่ใช้ทดสอบคุณสมบัติของแผ่นเวเฟอร์ จากที่ใช้หน่วยประมวลผลกลางแบบเก่า (Old CPU) มาเป็นหน่วยประมวลผลกลางแบบใหม่ (New CPU) โดยยังให้ผลการทดสอบที่ยังคงประสิทธิภาพที่ดีไว้เหมือนเดิม

2) การเปลี่ยนหน่วยประมวลผลกลางจากแบบเก่า (Old CPU) มาเป็นแบบใหม่ (New CPU) สามารถช่วยลดเวลาการทดสอบคุณสมบัติของแผ่นเวเฟอร์แต่ละแผ่นได้และยังสามารถเพิ่มผลผลิตที่ได้จากการทดสอบงานอีกด้วย

3) เพื่อเรียนรู้กระบวนการผลิต IC ในโรงงานการผลิตแบบของจริง ทั้งนี้เรายังสามารถเรียนรู้ได้ทั้งส่วนที่เป็นฮาร์ดแวร์ (เครื่องทดสอบแผ่นเวเฟอร์) และส่วนที่เป็นซอฟต์แวร์ (โปรแกรม)

1.3 สมมติฐานของการศึกษา

โปรแกรมที่สร้างขึ้นใหม่จากการพัฒนาโปรแกรมแบบเดิมที่ใช้หน่วยประมวลผลกลางแบบเก่า (Old CPU) เปลี่ยนมาเป็นหน่วยประมวลผลกลางแบบใหม่ (New CPU) สามารถนำมาทดสอบคุณสมบัติของแผ่นเวเฟอร์ได้อย่างมีประสิทธิภาพเทียบเท่ากับโปรแกรมดั้งเดิม

1.4 ขอบเขตของการวิจัย

นำโปรแกรมทดสอบคุณสมบัติของแผ่นเวเฟอร์ที่ใช้กับหน่วยประมวลผลกลางแบบเก่า (Old CPU) มาพัฒนาเป็นโปรแกรมที่ใช้หน่วยประมวลผลกลางแบบใหม่ (New CPU) โดยทำการปรับปรุงแก้ไขตัวโปรแกรมและทำการตรวจสอบผลโดยการเทียบผลการทดสอบของข้อมูลทั้งสองระบบโดยใช้เกณฑ์การทดสอบคุณสมบัติเดียวกัน

เริ่มปฏิบัติงานตั้งแต่วันที่ 7 สิงหาคม 2560 ถึง วันที่ 24 พฤศจิกายน 2560 โดยในช่วงเดือนมิถุนายนถึงกรกฎาคมเป็นช่วงของการฝึกงาน ในช่วงนั้นได้เรียนรู้งานเกี่ยวกับโครงการที่ได้รับมอบหมายแล้ว

รายละเอียด	สิงหาคม				กันยายน				ตุลาคม				พฤศจิกายน				
	สัปดาห์ที่																
	2	3	4	5	1	2	3	4	1	2	3	4	5	1	2	3	4
1. ทำการDebug test program ให้ผ่านเป็นbin1 หลังจากเปลี่ยนฮาร์ดแวร์ให้เป็นNew CPUแล้ว																	

รายละเอียด	สิงหาคม				กันยายน				ตุลาคม					พฤศจิกายน			
	สัปดาห์ที่																
	2	3	4	5	1	2	3	4	1	2	3	4	5	1	2	3	4
2.ทำtest program checker เพื่อแก้warning ในทุกๆtask พร้อมทั้งแก้warning ในhot spot detector ด้วย			←→														
3.ทำ spike check เพื่อตรวจสอบว่าpinไหนมีspikeบ้าง และทำการแก้spikeทุกๆpinด้วย				←→													
4.ทำstability check โดยการทดสอบงาน200ลูป พร้อมเก็บข้อมูลเพื่อทำการanalyzeต่อไป							←→										
5.ทำMSCเพื่อตรวจสอบ Reproducibilityและ Repeatabilityของทั้ง2ระบบ								←→									
6.ทำsite dependency ทำ 4 เจ็อนไซ เพื่อตรวจสอบ Reproducibilityและ Repeatability ทั้ง 4 site									←→								
7.ทำpilot run กับเวเฟอร์1แผ่น ทั้ง2ระบบ ตรวจสอบข้อมูลและ แก้ไข											←→						
8.นำเสนอโครงการในรูปแบบ power point ที่บริษัท															←→		
9.จัดทำรูปเล่มรายงาน																←→	

1.4.1 สถานที่ทำโครงการ

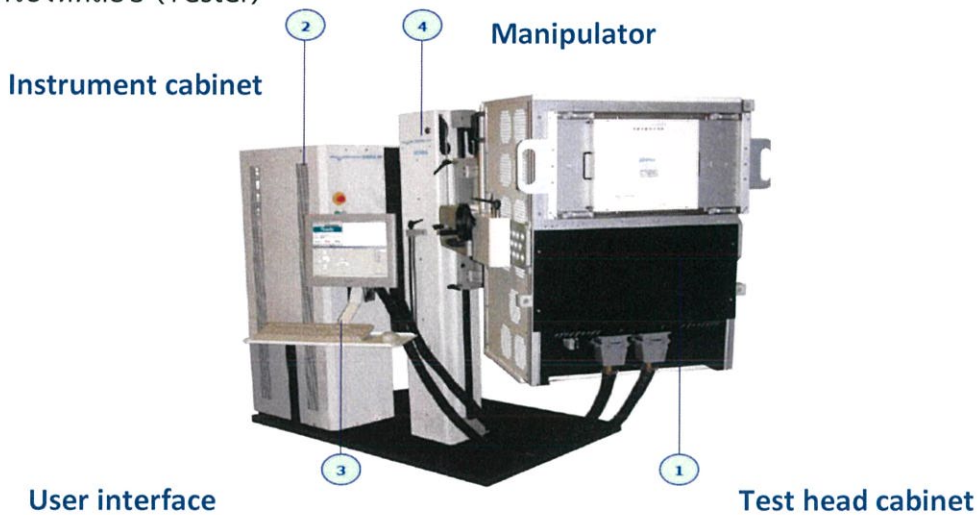
แผนก PE/TE ของบริษัทผลิตอุปกรณ์เซมิคอนดักเตอร์แห่งหนึ่ง

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

โดยภาพรวมของโครงการนี้ ประกอบไปด้วย 2 ส่วนใหญ่ๆ คือ ส่วนของเครื่องทดสอบและส่วน
ของโปรแกรมที่ใช้ทดสอบ

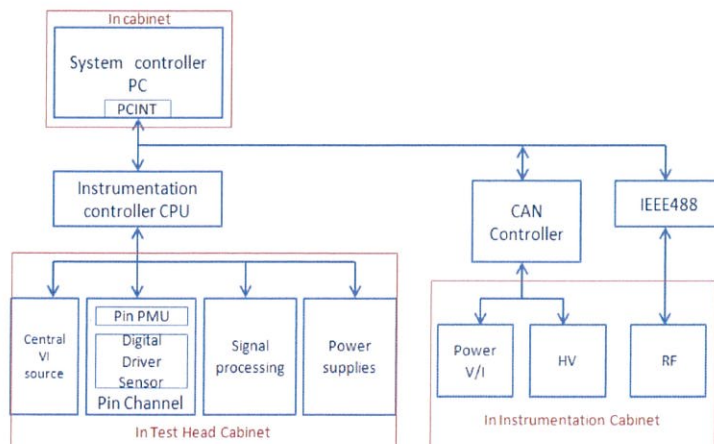
2.1 เครื่องทดสอบ (Tester)



รูปที่ 2.1 แสดงรูปของเครื่องที่ใช้ทดสอบ

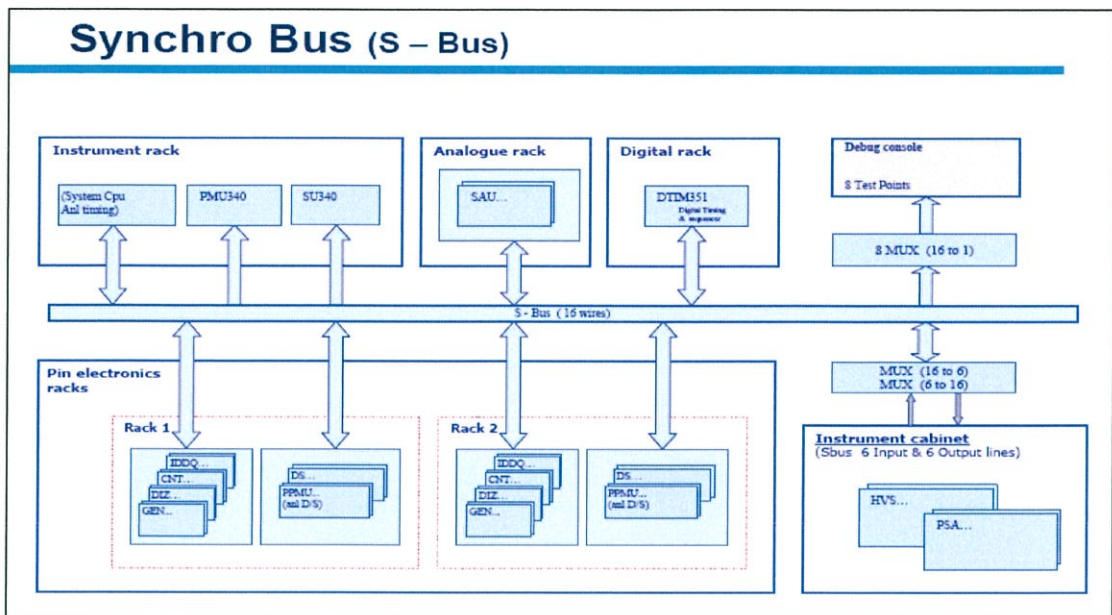
- Test head cabinet: ส่วนนี้จะเชื่อมกับ Handler เมื่อต้องการทดสอบงาน และในนี้จะประกอบไปด้วยบอร์ดต่างๆ ที่จำเป็นในการทำงานของเครื่อง
- Instrument cabinet: เป็นที่เก็บเครื่องมือต่างๆที่ใช้สำหรับการทดสอบงาน เช่น Power supply ที่ใช้สำหรับจ่ายแรงดันและกระแส, Multimeter เพื่อวัดค่าต่างๆที่ต้องการทราบ
- User interface: เป็นจอแสดงหน้าต่างต่างของ Control panel ของการทดสอบ
- Manipulator: เป็นที่ยึดของตัว Test head สามารถปรับระดับขึ้นลงได้

2.1.1 System Block Diagram



2.1.1.1 System buses ประกอบไปด้วย SBUS, ABUS และ MBUS

- SBUS หรือ Synchro Bus เป็น Bus ที่ประกอบไปด้วย 16 เส้น ใช้สำหรับเชื่อมสัญญาณส่งและสัญญาณรับในเส้นเดียวกันหรือจะใช้SBUSเชื่อมกับเส้นอื่นก็ได้โดยการเขียนโปรแกรม



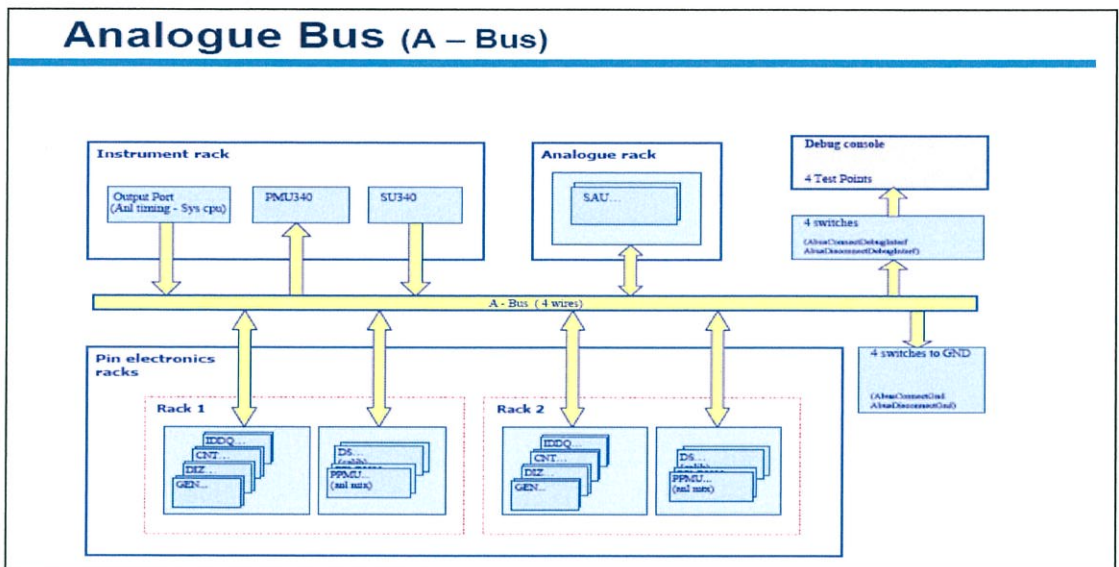
Line	Description
SBUS1	TPGMOFLAG2 (programmable) (*)
SBUS2	Programmable
SBUS3	Programmable
SBUS4	Programmable
SBUS5	Programmable
SBUS6	TPGMOFLAG1 (programmable) (*)
SBUS7	Programmable
SBUS8	WFGWORK
SBUS9	ANTIMSTART1
SBUS10	Programmable
SBUS11	Programmable
SBUS12	Programmable
SBUS13	APHTEST
SBUS14	APH1 (Analog phase 1)
SBUS15	APH2 (Analog phase 2)
SBUS16	APHM (Analog measurement phase)

Table 14

(*) SBUS1 and SBUS6 are mainly used by the System CPU.

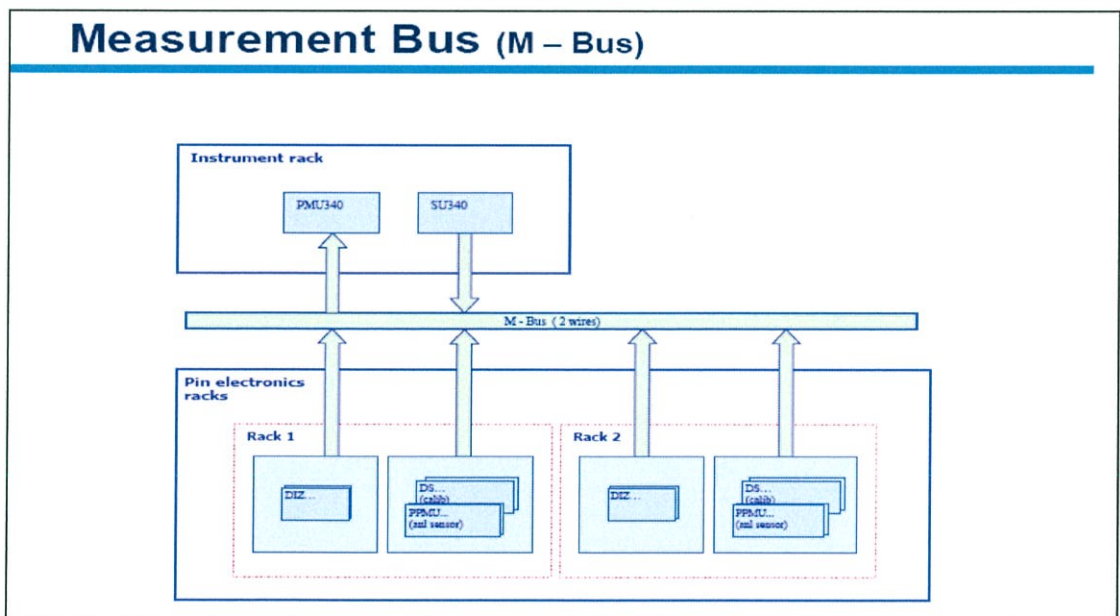
รูปที่ 2.2 แสดงโครงสร้างของ Synchro Bus และ ตารางแสดงว่าแต่ละSBUSทำอะไร

- ABUS หรือ Analogue Bus มีจำนวน 4 แถว ใช้สำหรับการวัดต่างๆไปและการเชื่อมต่อของระบบ เครื่องมืออนาล็อกทุกตัวสามารถเชื่อมต่อกับบัสนี้ เช่น บอร์ดPPMU, Central V/I Source



รูปที่ 2.3 แสดงโครงสร้างของAnalogue Bus (ABUS)

- MBUS หรือ Measurement Bus มีจำนวน 2 แฉก ใช้สำหรับการอ่านค่ากลับมาของแรงดันและกระแส เครื่องมือที่เป็น Digitizer, Pin Analog และ Pin Digital สามารถเชื่อมกับบัสนี้



รูปที่ 2.4 แสดงโครงสร้างของMeasurement Bus (MBUS)

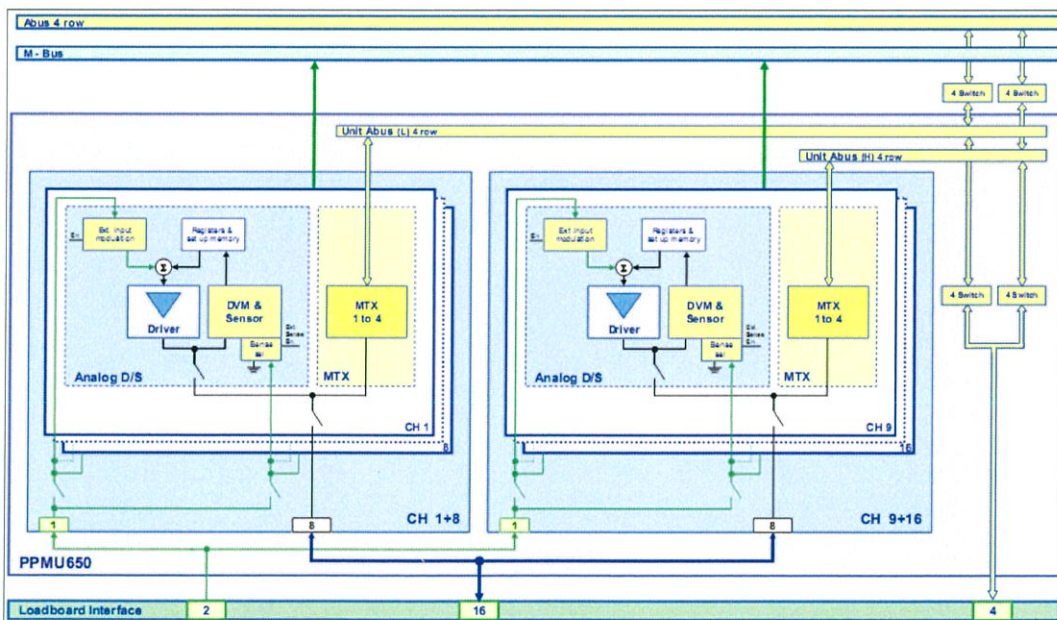
2.1.1.2 System Instrument ระบบของเครื่องทดสอบนั้นประกอบไปด้วยเครื่องมือต่างๆ ตามส่วนต่างๆของTester ดังนี้

	Instrument type	Name	Q.ty (up to)
Test head cabinet	Central V/I Source – DVM	SAU...	8
	General Purpose V/I Source – AWG	SU...	1
	Pin Analog	PPMU...	16
	Digital Driver/Sensors	DS...	16
	Power supplies	PSU...	1
	AC Source	GEN...	16 (*)
	Counter	CNT...	16 (*)
	Digitizer	DIZ...	16 (*)
	IDDQ Monitor	IDDQ...	16 (*)
	High Voltage Pin Analog	HPMU	8
Instrument cabinet	Power V/I Source	PSA...	4
	High Voltage V/I Source	HVS...	4
	RF Generator	RFG...	1

รูปที่ 2.5 แสดงเครื่องมือที่อยู่แต่ละส่วนของเครื่องทดสอบ

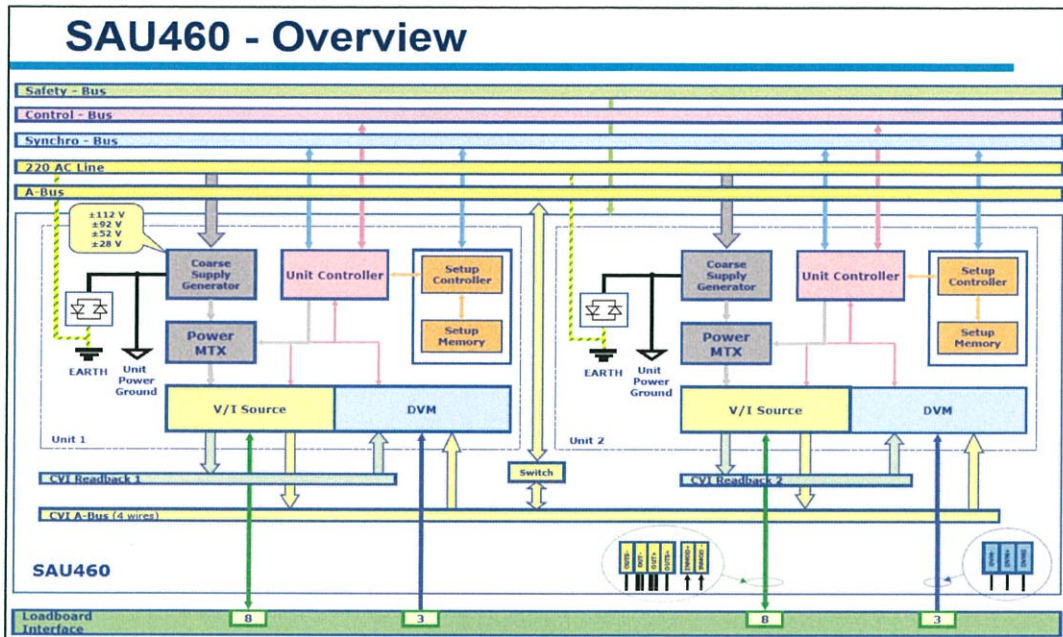
ในโครงงานนี้เราจะใช้เครื่องมืออยู่เป็นบางตัว คือ PPMU, SAU และ DS

- Analog pins(PPMU/P4MU) ประกอบไปด้วย 16 บอร์ด มี sampling time 40uS ใช้กับ SBUS ในการเชื่อมสัญญาณเข้าด้วยกัน มีการใช้ช่วงแรงดันช่วงเดียว คือ20V



รูปที่ 2.6 แสดงโครงสร้างของAnalog pins (PPMU/P4MU)

- SAU(CVI) จะใช้SAU460 มี2 แหล่งจ่ายแรงดันและกระแส รวมถึงมี2ส่วนของการวัดค่า ใช้กับแรงดันและกระแสสูงมากๆ



รูปที่ 2.7 แสดงโครงสร้างของSAU (CVI)

2.1.1.3 System Controller PC เป็นส่วนที่อยู่ใน cabinet ของเครื่องtester ส่วนนี้เป็นส่วนของหน่วยประมวลผลกลางหรือCentral Processing Unit (CPU) ซึ่งทำหน้าที่ ควบคุมการทำงานของอุปกรณ์ต่างๆในเครื่องtester ในที่นี้จะแบ่งเป็น 2 ประเภท คือ

- Old CPU Board(PCINT) เป็นบอร์ดCPUที่มี Controller CPU คือ YA5272 มีความเร็วในการประมวลผล66MHz



รูปที่ 2.8 แสดงรูปของบอร์ด Old CPU

- New CPU Board(PCXINT) เป็นบอร์ดCPUที่มี Controller CPU คือ YA5475 มีความเร็วในการประมวลผล266MHz



รูปที่ 2.9 แสดงรูปของบอร์ด New CPU

2.2 การเขียนโปรแกรมด้วยภาษา C++

โครงการนี้จะเขียนโปรแกรมด้วยภาษา C++ ซึ่งเป็นซอฟต์แวร์ที่เป็นภาษาคอมพิวเตอร์เพื่อวัตถุประสงค์ทั่วไป มีเครื่องมืออำนวยความสะดวกในการจัดการและเข้าถึงระดับหน่วยความจำ นอกจากนี้มันยังถูกนำไปใช้ในการเขียนโปรแกรมแบบต่างๆมากมาย เช่น โปรแกรมคอมพิวเตอร์ระบบฝังตัว (embedded) เว็บเซิร์ฟเวอร์ การพัฒนาเกม และแอปพลิเคชันที่ต้องการประสิทธิภาพอย่างสูง

ภาษาC++ เป็นภาษาที่ถูกออกแบบมาให้เขียนโปรแกรมระบบซึ่งมีประสิทธิภาพและความยืดหยุ่นในการออกแบบโปรแกรมสูง C++เป็นภาษาที่ต้องคอมไพล์ก่อนที่จะใช้งานซึ่งสามารถพัฒนาได้หลายแพลตฟอร์ม ได้รับการสนับสนุนจากองค์กรต่างๆ เช่น Free Software Foundation (FSF's GCC) LLVM Microsoft Intel และ IBM

2.2.1 โครงสร้างในภาษาC++

- Statement (คำสั่งการทำงาน) คือ คำสั่งการทำงานของโปรแกรมในภาษาC++ ในการเขียนโปรแกรมหนึ่งโปรแกรมนั้นโดยส่วนมากแล้วจะประกอบไปด้วยหลายคำสั่งเพื่อให้การทำงานเสร็จสิ้น เช่น `std::cout << "Hello World !";` นั่นคือคำสั่งการทำงานของโปรแกรมคำสั่งในภาษาC++นั้นจะจบด้วยเครื่องหมายเซมิโคลอน (;) ดังนั้นหลายคำสั่งอาจอยู่ในบรรทัดเดียวกันได้
- คอมเมนต์ เป็นส่วนของซอสโค้ดที่ไม่มีผลกับตัวโปรแกรม ช่วยให้โปรแกรมเมอร์สามารถอธิบายโปรแกรมของพวกเขาและช่วยตรวจสอบได้ง่ายในภายหลัง ในภาษาC++ มีสองทางเลือกในการคอมเมนต์

```
//          line comment
/* ..... */  block comment
```
- Using namespace std ภาษาC++มีไลบรารีมาตรฐานที่เรียกว่า namespace เพื่อใช้ฟังก์ชันในไลบรารีของ namespace std โดยไม่ต้องมี std:prefix เราจำเป็นต้องใช้คำสั่ง `using namespace std` อย่างเช่น ฟังก์ชันcout สามารถเรียกใช้ฟังก์ชันนี้ในทันที
- Whitespace คือตัวอักษรหรือเครื่องหมายที่ใช้แบ่งแยกคำสั่ง ในภาษาC++ นั้น whitespaceจะประกอบไปด้วย การเว้นวรรค Tab และการขึ้นบรรทัดใหม่ whitespace ที่เรียงต่อกันเป็นจำนวนมากนั้นไม่มีผลต่อการทำงานของโปรแกรมและคอมไพเลอร์แต่มันช่วย

ให้โปรแกรมเมอร์ทำได้ให้เป็นระเบียบและสามารถอ่านได้เข้าใจง่ายขึ้น แต่ Whitespace ยังคงต้องใช้ในบางที่ เช่น ระหว่างคำสั่งของภาษาC++ และชื่อของตัวแปร เป็นต้น

2.2.2 คำสั่งควบคุมการทำงานของโปรแกรม

- คำสั่ง If เป็นคำสั่งที่เป็นพื้นฐานที่สุดสำหรับควบคุมการทำงานในภาษาC++ คำสั่ง If ถูกใช้เพื่อควบคุมโปรแกรมกับเงื่อนไขที่กำหนด โค้ดในบล็อกของคำสั่ง If จะทำงานเมื่อเงื่อนไขตรงหรือเป็นจริง เรามักจะใช้คำสั่ง If ในกรณีที่ต้องทำงานภายใต้เงื่อนไขบางอย่าง รูปแบบในการใช้งานคำสั่ง If ในภาษาC++ เป็นดังนี้

```
if (expression) {  
    // statements  
}
```

รูปที่ 2.10 แสดงรูปแบบการใช้คำสั่ง If ในภาษาC++

- คำสั่ง If else คล้ายกับคำสั่ง If แต่จะใช้สำหรับสร้างเงื่อนไขแบบหลายทางเลือก และมันจะต้องเริ่มต้นที่คำสั่ง If เสมอ และนอกจากนี้ยังมีคำสั่ง else clause เพื่อทำในเงื่อนไขที่นอกเหนือจากเงื่อนไขอื่นทั้งหมด ดังเช่นในตัวอย่างต่อไปนี้

```
int m = -1;  
if (n < 0) {  
    cout << "Negative number."  
}  
else if (n > 0) {  
    cout << "Positive number."  
}  
else {  
    cout << "Zero number";  
}
```

รูปที่ 2.11 แสดงตัวอย่างการใช้คำสั่ง If else ในภาษาC++

จากตัวอย่าง คำสั่ง If สามารถมีเงื่อนไขได้หลายอันโดยการใช้ **else if()** สำหรับสร้างเงื่อนไขที่สองเป็นต้นไป และในเงื่อนไขสุดท้ายคือ **else** ซึ่งมันจะทำงานเมื่อไม่ตรงกับเงื่อนไขใดๆ ก่อนหน้า ในโค้ด เรามีตัวแปร m ซึ่งมีข้อมูลเป็นแบบ integer โปรแกรมของเรานั้นจะตรวจสอบว่า m เป็นจำนวนเต็มบวก เต็มลบ หรือศูนย์

- คำสั่ง while loop เป็นลูปที่ง่ายและพื้นฐานที่สุด เราใช้คำสั่ง loop ในการวนซ้ำให้โปรแกรมสามารถทำงานเดิมซ้ำๆได้ โดยมีรูปแบบการใช้งานคือ

```
while (expression) {  
    statements  
}
```

รูปที่ 2.12 แสดงรูปแบบของการใช้คำสั่ง while loop

คำสั่ง while loop ใช้เพื่อสั่งโค้ดของโปรแกรมในขณะที่ expression เป็นจริง และมันจะสิ้นสุดการทำงานเมื่อ expression เป็นเท็จและออกจากwhile loop เพื่อทำคำสั่งต่อไป

- คำสั่ง for loop เป็นลูปที่มีการวนรอบเป็นจำนวนที่แน่นอน for loop เป็นลูปที่สามารถวนรอบตามตัวเลขที่กำหนดได้ มันทำงานเหมือน while-loop มันจะวนซ้ำจนกว่า expression จะเป็นเท็จ นอกจากนั้น เรายังสามารถประกาศตัวแปรเริ่มต้น สร้าง expression เพิ่มและลดค่าก่อนที่ลูปจะเริ่มรูปแบบของคำสั่งนี้ คือ

```
for (initialize; condition; increase) {  
    statements  
}
```

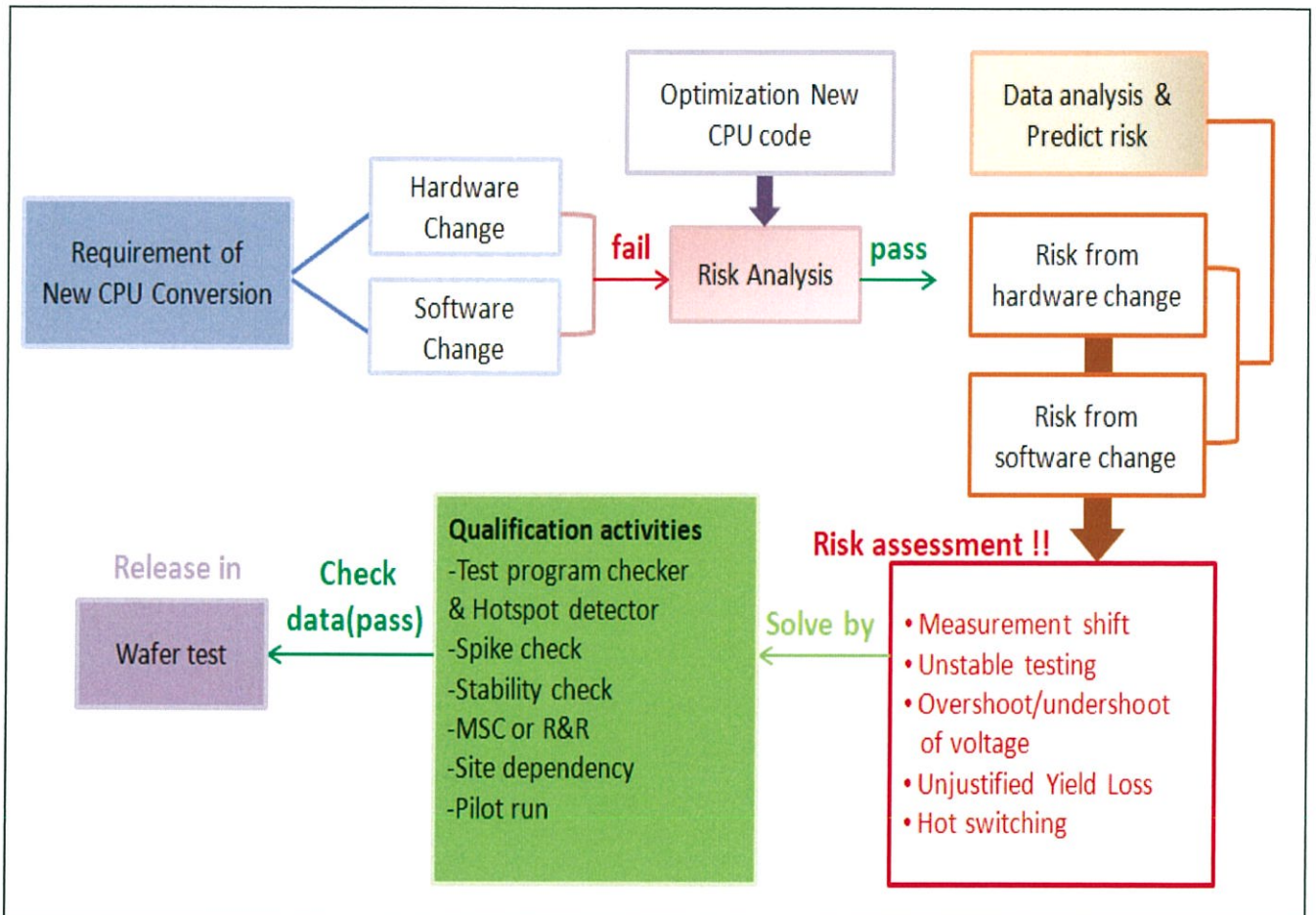
รูปที่ 2.13 แสดงรูปแบบการใช้คำสั่ง for loop

บทที่ 3

วิธีการดำเนินงาน

ในการจัดทำโครงการเรื่อง Integrated Circuit Test Time Reduction by New CPU Conversion นี้ผู้จัดทำมีวิธีการดำเนินงานตามแผนผังดังต่อไปนี้

3.1 ขั้นตอนการดำเนินโครงการ

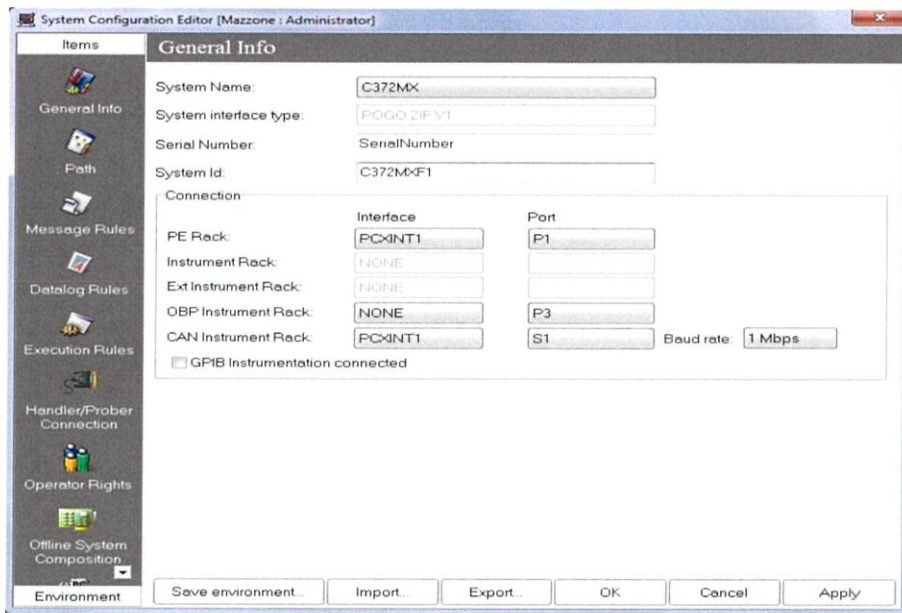


รูปที่ 3.1 แสดงภาพรวมของการดำเนินโครงการ

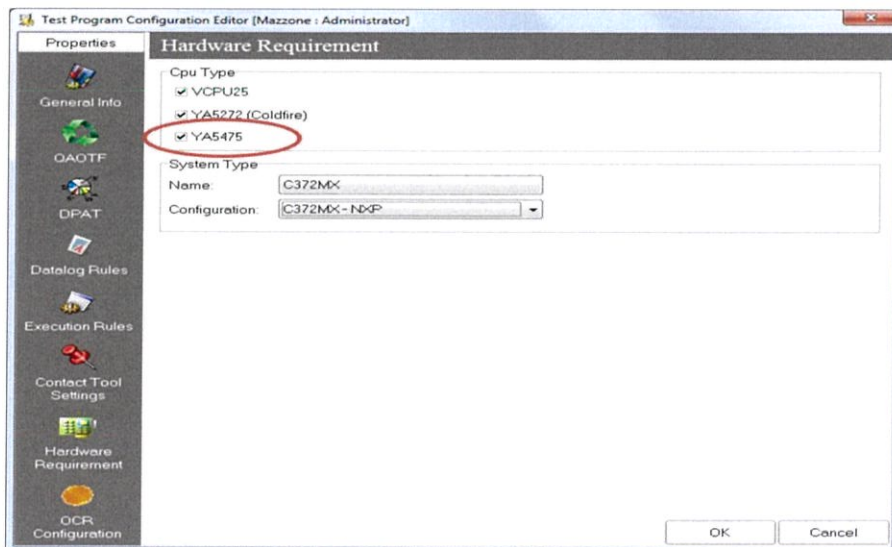
ในส่วนของขั้นตอนการดำเนินโครงการนั้น เราจะทำการพัฒนาโปรแกรมที่ใช้หน่วยประมวลผลกลางแบบเก่า(Old CPU) มาเป็นโปรแกรมที่ใช้หน่วยประมวลผลกลางแบบใหม่ (New CPU) วิธีการนี้เราเรียกว่า "New CPU Conversion Test Program" ซึ่งโปรแกรมที่สร้างขึ้นใหม่นั้นสามารถใช้งานได้และเป็นไปตามจุดประสงค์ที่ต้องการทุกประการ โดยมีขั้นตอนตั้งแต่เริ่มแรกจนถึงสิ้นสุด ดังนี้

3.1.1 Requirement of New CPU Conversion

การที่เราจะทำการสร้างโปรแกรมที่ใช้หน่วยประมวลผลกลางแบบใหม่(New CPU)นั้น เราต้องทำการเปลี่ยนฮาร์ดแวร์ให้กับเครื่องทดสอบก่อน ซึ่งเราจะทำการเปลี่ยนจากOld CPU Board(PCINT) ที่มี Controller CPU คือYA5272 เป็นNew CPU Board(PCXINT) ที่มี Controller CPU คือ YA5475 เมื่อเปลี่ยนบอร์ดเสร็จต้องทำการเปลี่ยนSystem configuration ด้วย คือเปลี่ยนจาก PCINT เป็น PCXINT1 ดังรูป และเปลี่ยน Hardware requirement เป็น YA5475 ด้วยดังรูป



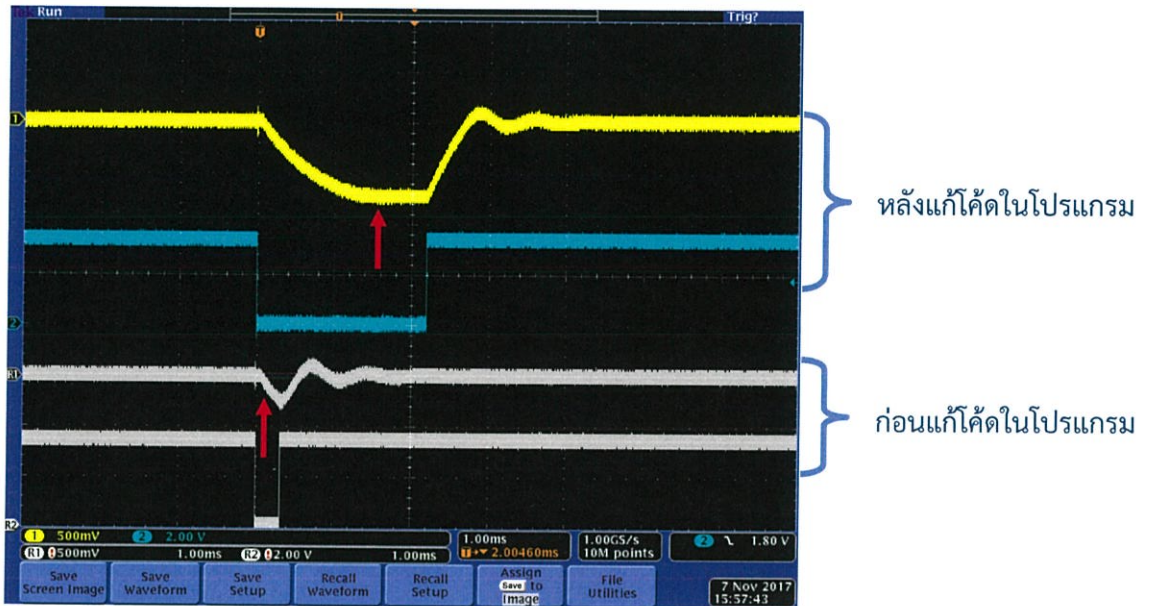
รูปที่ 3.2 แสดงหน้าต่างของSystem configuration



รูปที่ 3.3 แสดงหน้าต่างของHardware requirement

3.1.2 Risk analysis

เมื่อทำการเปลี่ยนฮาร์ดแวร์เสร็จพบว่าเมื่อนำโปรแกรมเก่าไปทำการทดสอบที่ไลน์การผลิตพบว่าผลของการทดสอบเกิดการเฟลขึ้น ดังนั้นเราจึงต้องทำการแก้ไขโค้ดหรือ Optimization New CPU code เกิดการเฟลที่ task ของ contact short สามารถดูสัญญาณได้ดังนี้



รูปที่ 3.4 รูปสัญญาณของการวัดแรงดันใน Task contact short

จากสัญญาณดังกล่าวจะเห็นว่าสัญญาณก่อนที่จะแก้ไขโค้ดนั้นมีการstrobe หรือวัดตรงจุดที่ไม่เหมาะสม ไม่คงที่ ทำให้ค่าที่อ่านได้นั้นไม่อยู่ในลิมิตทำให้เกิดการเฟลขึ้น จึงทำการเพิ่มเวลาการเริ่มstrobeให้มากขึ้น ดังสัญญาณหลังแก้ไขโค้ดในโปรแกรม ซึ่งมีการวัดตรงจุดที่เหมาะสมแล้ว เพราะเป็นจุดที่อ่านค่าได้คงที่และถูกต้อง

3.1.3 Data analysis & Predict risk

หลังจากทำการ Optimization New CPU codeจนผ่านเป็นbin1 แล้ว ก็ลองทำการทดสอบแต่ละtask ในทุกๆsiteดูพบว่าค่าที่อ่านออกมาในแต่ละsiteนั้น ไม่คงที่ ไม่ถูกต้องถึงจะอยู่ในลิมิตก็ตาม ผลการทดสอบนั้นเกิดจากการที่เปลี่ยนCPU ซึ่งNew CPU นั้นมีการประมวลผลที่เร็วไป จึงมีการ Risk assessment ได้ดังนี้

- Measurement shift : การที่ค่าที่วัดได้คลาดเคลื่อนระหว่างsite
- Unstable testing : ผลการทดสอบนั้นไม่คงที่ในทุกๆsite

- Overshoot/undershoot of voltage : เกิดการกระเพื่อมของแรงดันเกินสเป็คใน datasheet ซึ่งมีความเสี่ยงที่จะทำให้งานเสียหายได้
- Unjustified Yield Loss : ที่โปรแกรมใหม่เกิดการLoss ของ Yield(ค่าเปอร์เซ็นต์บ่งบอกประสิทธิภาพในการทดสอบคุณสมบัติแผ่นเวเฟอร์ว่าได้Dieที่ผ่านเกณฑ์คุณสมบัติมากน้อยเพียงใด)
- Hot switching : เกิดแรงดันที่Relay ก่อนสับสวิตซ์ปิดและหลังสับสวิตซ์เกินกว่าที่อุปกรณ์จะทนได้ อาจทำให้เกิดความเสียหายของอุปกรณ์ได้

3.1.4 Qualification activities

เป็น activities ที่จำเป็นหลังจากที่เราพัฒนาโปรแกรมเก่าแล้ว ทำเพื่อเปรียบเทียบระหว่าง reference test program(โปรแกรมเก่า) กับ target test program(โปรแกรมใหม่) และสามารถระบุปัญหาที่เกิดขึ้นได้ว่าเกิดจากตัวโปรแกรมเองหรือเป็นที่ Hardware equipment ถ้าเกิดจากโปรแกรมต้องทำการแก้ไข ซึ่งQualification activitiesนั้นสามารถแบ่งออกเป็น 6 อย่าง ดังต่อไปนี้

- Test program checker & Hot spot detector : สามารถแก้ปัญหาเรื่อง Hot switching
- Spike check : สามารถแก้ปัญหาเรื่อง Overshoot/undershoot ของ voltage ได้ซึ่งแรงดันที่maximumและminimumนั้นอาจทำให้งานที่ทดสอบเกิดความเสียหายได้
- Stability check : สามารถแก้ปัญหาเกี่ยวกับ Unstable testing ได้
- MSC หรือ R&R : สามารถแก้ปัญหาเกี่ยวกับ Measurement shift ได้ เมื่อเกิดค่าที่อ่านได้คลาดเคลื่อนทุกๆsite
- Site dependency : สามารถแก้ปัญหาเกี่ยวกับ Measurement shift ได้
- Pilot run : สามารถแก้ปัญหาเรื่อง Unjustified Yield Loss ได้

3.1.4.1 Test program checker & Hot spot detector

Test program checker นั้นทำเพื่อตรวจจับสิ่งที่ผิดปกติของตัวโปรแกรม เช่น การเรียงลำดับของการตั้งค่าก่อนจะเริ่มทดสอบ ที่ถูกต้องตามลำดับคือ connect-set-enable, set-disable-disconnect ซึ่งก็ได้แก้errorทั้งหมดเรียบร้อยแล้ว ได้ผลดังนี้

```

Task Pincontinuity: Pass
Task ContactResistance: Pass

Warning: Test program checker!
>> Task: Maxrating
>> Test: 12009
>> Instruction: PpsuSourceSet
Voltage range change to a range greater or equal than 40V!
Check for minimum required settling time as specified in the Technical Info

Warning: Test program checker!
>> Task: Maxrating
>> Test: 12010
>> Instruction: PpsuSourceSet
Voltage range change to a range greater or equal than 40V!
Check for minimum required settling time as specified in the Technical Info

Warning: Test program checker!
>> Task: Maxrating
>> Test: 12011
>> Instruction: PpsuSourceSet
Voltage range change to a range greater or equal than 40V!
Check for minimum required settling time as specified in the Technical Info

Warning: Test program checker!
>> Task: Maxrating
>> Test: 12011
>> Instruction: PpsuSourceSet
Voltage range change to a range greater or equal than 40V!
Check for minimum required settling time as specified in the Technical Info

Task Maxrating: Pass
Task supplycurrent: Pass
Task Scantest: Pass
Task Voltagestress: Pass
Task Iocquesting: Pass
Task DeUnderVoltage: Pass
Task DeTXD: Pass
Task DeSTBN: Pass
Task DeRXD: Pass

Warning: Test program checker!
>> Task: DeCANTX
>> Test: 271110
>> Instruction: PpsuSourceSet
Voltage range change to a range greater or equal than 40V!
Check for minimum required settling time as specified in the Technical Info

Warning: Test program checker!
>> Task: DeCANTX
>> Test: 271110
>> Instruction: PpsuSourceSet
Voltage range change to a range greater or equal than 40V!
Check for minimum required settling time as specified in the Technical Info

Warning: Test program checker!
>> Task: DeCANTX
>> Test: 271341
>> Instruction: PpsuSourceSet
Voltage range change to a range greater or equal than 40V!
Check for minimum required settling time as specified in the Technical Info

Warning: Test program checker!
>> Task: DeCANTX
>> Test: 271410
>> Instruction: PpsuSourceSet
Voltage range change to a range greater or equal than 40V!
Check for minimum required settling time as specified in the Technical Info

Task DeCANTX: Pass
Task DeCANRXTHR: Pass
Task DeCANRXGEN: Pass
Task Timingmeasurement: Pass
Task Temperatureprotection: Pass
Task IDbit: Pass
Task PostContactShort: Pass
Task Movinglimits: Pass
Task Exec_Info: Pass

```

รูปที่ 3.4 แสดงผลหลังจากทำTest program checker แล้ว

Hot spot detector นั้นเป็นการตรวจสอบว่ามี Hot switching หรือไม่ซึ่งมันก็คือแรงดันที่เกิดจากแรงดันหลังจากที่สับสวิตช์relay กับแรงดันก่อนที่จะสับสวิตช์relayมาลบกัน ซึ่งไม่ควรเกินกว่าที่ใช้ทไวท์ให้กับโปรแกรม ซึ่งในโปรแกรมได้ตั้งค่าไว้ว่าห้ามเกิน3V (ตั้งตามSPEA User Group Criteria) เมื่อแก้ไขเรียบร้อยแล้วได้ผลดังนี้

```

-----
WARNING:
P4MD HOT SPOT DETECTOR is enabled
Measure and Test time could be affect by this function
HOT SPOT SETTINGS:
Mode = V_Mode
I Range = P4R4uA
Value (A) = 5.000E-08
Th+/Th- value (V) = 5.00
Delay(us) = 500.00
Width(us) = 100.00
Task Pincontinuity: Pass
Task ContactResistance: Pass
Task Maxrating: Pass
Task supplycurrent: Pass
Task Scantest: Pass
Task Voltagestress: Pass
Task Iocquesting: Pass
Task DeUnderVoltage: Pass
Task DeTXD: Pass
Task DeSTBN: Pass
Task DeRXD: Pass
Task DeCANTX: Pass
Task DeCANRXTHR: Pass
Task DeCANRXGEN: Pass
Task Timingmeasurement: Pass
Task Temperatureprotection: Pass
Task IDbit: Pass
Task PostContactShort: Pass
Task Movinglimits: Pass
Task Exec_Info: Pass

```

รูปที่ 3.5 แสดงผลหลังจากทำHot spot detector แล้ว

3.1.4.2 Spike check

เป็นการตรวจสอบว่ามีแรงดันที่ไม่ต้องการหรือไม่ เมื่อวัดที่ maximum และ minimum voltage เราจะได้จากรูปสัญญาณที่วัดออกมาจากเครื่องออสซิลอสโคป ซึ่งแรงดันดังกล่าวถ้าเราไม่ทำการแก้ไขสัญญาณspikeให้หายไปก็จะเป็นความเสี่ยงให้งานที่ทำการทดสอบเกิดความเสียหายได้ ซึ่งสัญญาณspike อาจเกิดจากการป้อนแรงดันและกระแสที่ไม่เหมาะสม, New CPU มีการประมวลผลที่เร็วเกินไปทำให้มีการวัดค่าในตำแหน่งที่ไม่เหมาะสม, มีการป้อนแรงดันแบบทันทีควรใส่ wait time เพื่อลดสัญญาณspike

จากการวัดสัญญาณแต่ละpin ที่ maximum และ minimum voltage แล้วนั้นพบว่า มีค่าแรงดันที่pinต่างๆ ดังตารางต่อไปนี้

The table below shows maximum and minimum voltages of each pin

Pin no.	Pin name	Max rating in spec		Site1		Site2		Site3		Site4	
		Max(V)	Min(V)	Max(V)	Min(V)	Max(V)	Min(V)	Max(V)	Min(V)	Max(V)	Min(V)
1		7	-0.3	5	-0.652	4.98	-0.668	4.98	-0.668	5	-0.656
2		7	-0.3	7.36	-0.692	7.28	-0.712	7.24	-0.716	7.24	-0.7
6		7	-0.3	9.04	-1.4	8.92	-1.52	9	-1.42	8.88	-1.46
7		7	-0.3	9.04	-1.42	8.92	-1.58	9	-1.48	8.88	-1.5
8		7	-0.3	8.12	-0.696	8.16	-0.716	8.08	-0.716	8.04	-0.704
9		7	-0.3	9	-1.72	8.84	-1.74	9	-1.8	8.88	-1.74
10		7	-0.3	9	-1.66	8.88	-1.7	9.04	-1.72	8.84	-1.68
11		7	-0.3	8.08	-0.664	8.04	-0.684	8	-0.688	8	-0.672
15		7	-0.3	7.32	-0.692	7.24	-0.712	7.24	-0.716	7.24	-0.7
16		7	-0.3	8.09	-0.664	8.07	-0.684	8.08	-0.688	8.05	-0.672
17		7	-0.3	5.64	-0.716	5.58	-0.736	5.62	-0.736	5.58	-0.728
18		58	-58	57.8	-35.2	57.6	-35.4	57.6	-35.2	57.2	-35.4
19		58	-58	57.8	-35.2	57.6	-35.4	57.6	-35.2	57.2	-35.4
20		58	-58	57.8	-35	57.8	-35.4	57.6	-35.2	57.2	-35.4
21		58	-58	57.8	-35	57.6	-35.2	57.6	-35.2	57.2	-35.4
22		7	-0.3	8.08	-0.664	8.04	-0.684	8.06	-0.684	8.04	-0.672
23		7	-0.3	8.08	-19.2	8.08	-19.3	8.04	-19.3	8	-19.3
24		58	-58	58	-34.6	57.8	-35.2	57.6	-35	57.2	-35
25		58	-58	57.8	-34.6	57.6	-35.2	57.6	-35	57.2	-35
26		58	-58	57.8	-34.6	57.6	-35.2	57.6	-35	57.2	-35
27		58	-58	57.8	-34.6	57.8	-35.2	57.6	-35	57.2	-35
28		7	-0.3	14	-1.41	13.9	-1.45	13.9	-1.46	13.9	-1.44

Red color value means over maximum specification voltage.
Blue color value means lower than minimum specification voltage.

ตารางที่ 3.1 แสดงค่าแรงดันที่วัดได้ของแต่ละ pin ที่ maximum และ minimum voltage

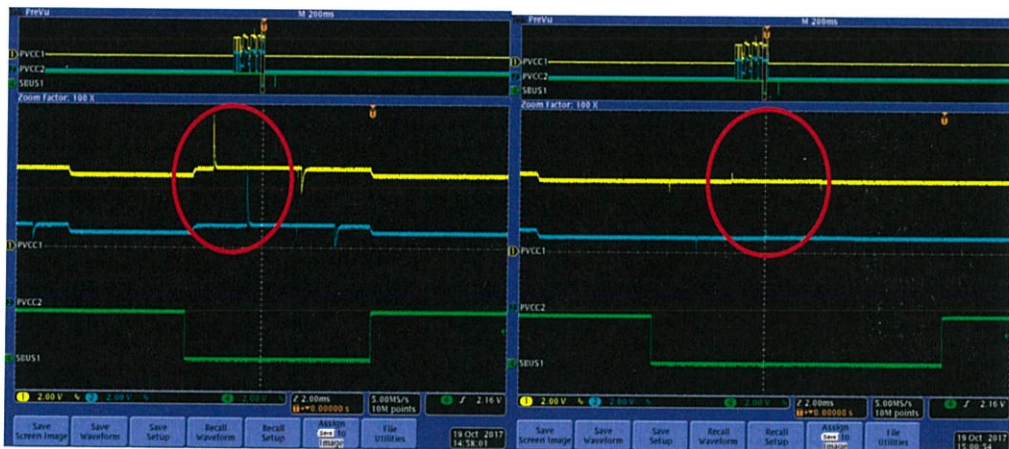
Spike analysis

จะทำการจับสัญญาณของpinต่างๆ เพื่อตรวจสอบว่าpinไหนมีspikeบ้าง โดยสัญญาณของspikeนั้นจะมีทั้งเป็นmaximum และ minimum voltage ลักษณะของสัญญาณspikeจะเป็นสัญญาณที่มีความแหลม เป็นการกระเพื่อมของแรงดันเกินที่ควรจะเป็นหรือเกินที่ตัวงานจะทนได้

การทำการตรวจสอบspikeมีความจำเป็นมากก่อนที่จะทำขั้นตอนต่อไป ควรทำขั้นตอนนี้ก่อน เพื่อลดความเสี่ยงที่ตัวงานจะได้รับ ความเสียหายหลังจากที่มีการป้อนไฟเข้าที่ตัวงานเพื่อทำการทดสอบ

การแก้สัญญาณspike ที่pin ต่างๆ

- Positive spikes(at maximum voltages)



ก่อนแก้

หลังแก้

เกิดสัญญาณspikeทั้ง 2 pin แก้ไขโปรแกรมโดยการเปลี่ยนช่วงของกระแสที่บอร์ดP4mu จาก40mAเป็น400uA หลังจากแก้เสร็จสัญญาณspikeก็หายไป



ก่อนแก้

หลังแก้

เกิดสัญญาณspikeทั้ง 2 pin แก๊ซโปรแกรมโดยการเพิ่มคำสั่งP4muSourceSet ในเทสโปรแกรม



ก่อนแก้

หลังแก้

เกิดสัญญาณspikeทั้ง 2 pin แก๊ซโปรแกรมโดยการเปลี่ยนตำแหน่งการdisable ให้มาอยู่บรรทัดสุดท้ายและเปลี่ยนช่วงของกระแสที่บอร์ดP4mu จาก40mAเป็น4mA หลังจากแก้เสร็จสัญญาณspikeก็หายไป

** เมื่อแก้สัญญาณ spike ที่ maximum voltage เสร็จ ก็ตรวจสอบที่ maximum voltage ของทุก pin อีกครั้ง พบว่าไม่มีสัญญาณ spike ที่ maximum voltage แล้ว

- Negative spikes(at minimum voltages)



ก่อนแก้

หลังแก้

เกิดสัญญาณspikeทั้ง 2 pin แก้วโปรแกรมโดยการสลับลำดับระหว่าง connectกับset จาก set-connect เป็น connect-set และย้ายการ connection ของrelay capacitor 10uF แก้วเสร็จสัญญาณspikeก็หายไป



ก่อนแก้

หลังแก้

เกิดสัญญาณspikeทั้ง 2 pin แก้วโปรแกรมโดยการค่อยๆdisable แรงดันจากที่ตอนแรก disableแรงดันทันที โดยการใส่wait time แล้วทำการstep down แรงดันลงมาเรื่อยๆ



ก่อนแก้

หลังแก้

เกิดสัญญาณspikeทั้ง 2 pin แก๊ซโปรแกรมโดยการค่อยๆstep upแรงดัน โดยการใส่ wait time



ก่อนแก้

หลังแก้

เกิดสัญญาณspikeทั้ง 2 pin แก๊ซโปรแกรมโดยการย้ายการdisconnection ของ relay capacitor 10uFไปไว้บรรทัดสุดท้าย

** เมื่อแก้สัญญาณ spike ที่ minimum voltage เสร็จ ก็ตรวจสอบที่ minimum voltage ของทุก pin อีกครั้ง พบว่าไม่มีสัญญาณ spikeที่ minimum voltageแล้ว

3.1.4.3 Stability check

เป็นการตรวจสอบว่าค่าที่ได้จากการทดสอบนั้นมีความคงที่หรือไม่ ซึ่งควรมีค่าใกล้เคียงกันในทุกๆsite และค่าที่อ่านได้ควรใกล้เคียงกับโปรแกรมเก่าที่ใช้กับหน่วยประมวลผลกลาง แบบเก่า (Old CPU) วิธีการทำก็คือ นำงานมาที่เป็นงานดีไม่เสีย มาทดสอบทั้งหมด200ลูป ทำเช่นนี้ทั้งOld CPUและNew CPU พร้อมทั้งเก็บข้อมูลมาเพื่อมาวิเคราะห์ข้อมูลผ่านเครื่องมือหนึ่งที่จะคำนวณค่าทางสถิติให้ เพื่อตรวจสอบว่าเทสไหนมีค่า Cp ต่ำกว่า 12.3 บ้าง ซึ่งเราควรจะแก้ไขให้ไม่เป็นเช่นนั้น หรือสามารถระบุปัญหาได้ว่าสาเหตุไม่ได้เกิดจากตัวโปรแกรม

ผลการทดสอบของแต่ละtest พบว่าtestที่มีค่า $Cp < 12.3$ จะไม่ได้เกิดจากตัวโปรแกรมใหม่ที่สร้างขึ้น เนื่องจากค่า Cp ที่ได้จากการทำNew CPU Test program มีค่าใกล้เคียงกับ Old CPU Test program จึงถือว่า comparable กัน ไม่เป็นปัญหาแต่อย่างใด

3.1.4.4 MSC (Measurement System Comparison)

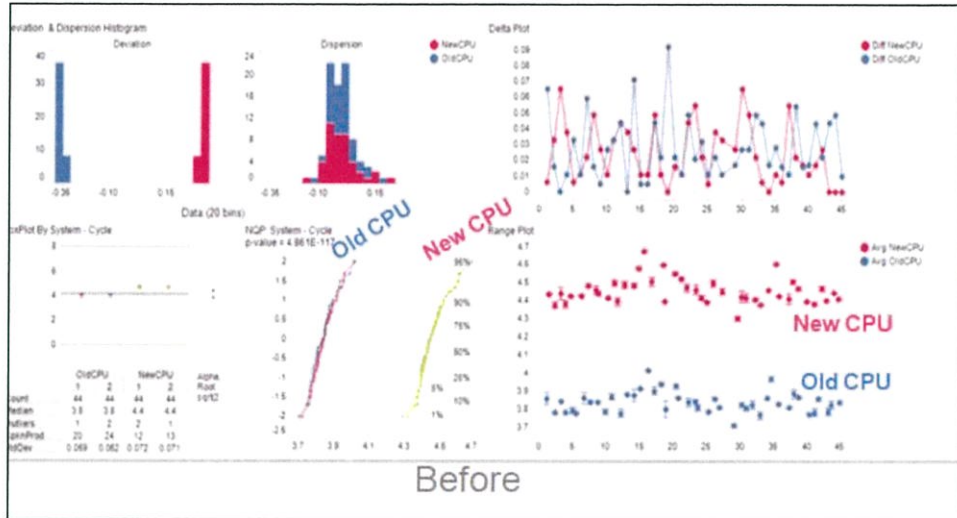
MSC เป็นการเปรียบเทียบสองระบบระหว่างตัวโปรแกรมเก่ากับโปรแกรมใหม่ ค่าที่อ่านได้ควรมีค่าใกล้เคียงกัน วิธีการทำ คือ นำงานที่จะทดสอบมา 45 ตัว ทดสอบที่siteเดียวจนครบ 45 ตัว ทดสอบทั้งที่Old CPU และ New CPU และเก็บข้อมูลมาเพื่อมาวิเคราะห์ค่าทางสถิติที่จำเป็นทางเครื่องมือหนึ่ง ค่าสถิติที่นำมาวิเคราะห์ คือ R&R ควรมีค่าน้อยกว่า10%, Reproducibility ควรมีค่าน้อยกว่า 7%, Repeatability ควรมีค่าน้อยกว่า 7%

ผลการทดสอบนั้นพบว่ามียางtestที่ไม่เป็นไปตามเกณฑ์ เราจึงต้องมาวิเคราะห์ว่าเกิดจากตัวโปรแกรมที่สร้างขึ้นหรือไม่ ดังนี้

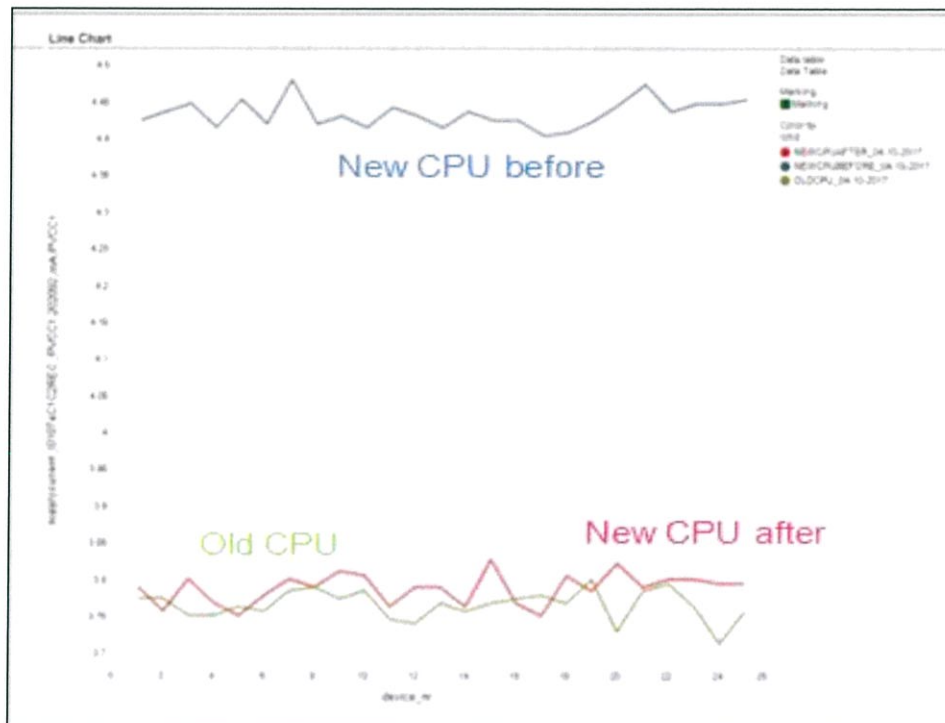
Q2	R&R	Reproducibility	Repeatability	Repeat - NewCP	Repeat - OldCP
1.2205	100	100	6.628727335	6.488001471	6.766527107
1.3855	100	100	4.710738544	4.817530364	4.601468944
4.151	28.061	28.02229249	1.472524158	1.372428143	1.566236185
55.2725	27.3822	27.38102427	0.258069155	0.294834405	0.215109393
61.382	27.3392	27.33366201	0.549927003	0.56785578	0.531393668
4.0645	25.5093	25.49360799	0.893259142	0.948650521	0.834197805
52.25	24.5794	24.57474588	0.479862613	0.543103592	0.406908765
46.1645	24.5435	24.54202161	0.2701876	0.326938356	0.197772571
10.536	22.5755	22.56296624	0.753421495	0.775826956	0.730328989
45.769	22.4256	22.42130916	0.440434579	0.375289458	0.497114734
52.0115	22.3779	22.37316332	0.461756392	0.421814688	0.498508074
144.714	21.3803	21.05466213	3.717150655	2.833252359	4.427990409
-0.188	15.8171	15.54820573	2.903881451	3.852714308	1.421846484
0.18	15.2049	14.93491247	2.852835473	3.745174723	1.500335552
150.786	13.0839	12.1381303	4.884148428	4.773408785	4.992432303
-0.144	11.1011	11.0998303	0.168630779	0.238479934	0
0.2415	9.81053	9.68145496	1.58619221	0.240744035	2.230258676

รูปที่ 3.6 แสดงผลการทดสอบบาง test ที่ไม่เป็นไปตามเกณฑ์ที่กำหนดไว้

MSC Solution



รูปที่ 3.7 กราฟการกระจายตัวของข้อมูลที่ทำ MSC ก่อนแก้โปรแกรม

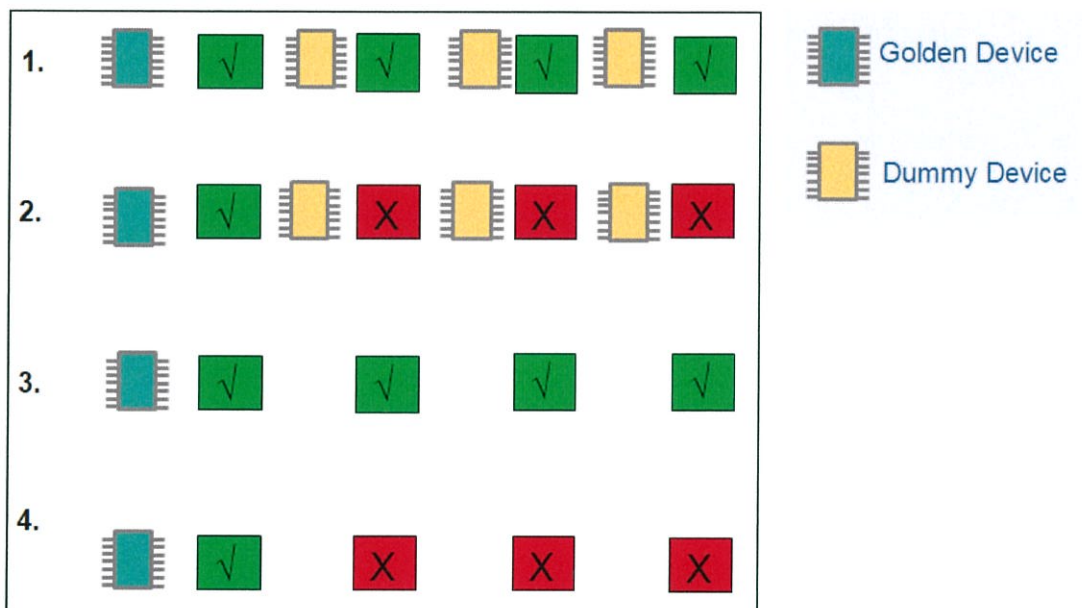


รูปที่ 3.8 กราฟการกระจายตัวของข้อมูลที่ทำ MSC หลังแก้โปรแกรม

จากกราฟกระจายตัวของข้อมูลก่อนแก้ไขโปรแกรมจะสังเกตเห็นว่าค่าที่อ่านได้ของ New CPU และ Old CPU มีการ shift กันมาก มีค่า Reproducibility > 7% แก้ไขโดยการใส่ wait time หลังจากทำการ set voltage แล้วทำการวัดค่า สามารถแก้ไขเรื่อง Reproducibility > 7% ได้ สังเกตได้จากกราฟกระจายตัวของข้อมูลหลังแก้ไขโปรแกรม

3.1.4.5 Site dependency

เป็น activity ที่ตรวจสอบทุกๆ site ว่าค่าที่วัดได้คงที่และมีการ shift กันระหว่าง site หรือไม่ โดยเราจะพิจารณาค่าสถิติเหมือน MSC คือ ค่า R&R, Reproducibility และ Repeatability ควร มีค่าน้อยกว่า 7% ในที่นี้จะทำ 4 เงื่อนไข ดังนี้



รูปที่ 3.9 แสดงเงื่อนไขที่ใช้ในการทำ site dependency

ในการทำ site dependency จะทำทั้งหมด 4 เงื่อนไข โดยมีรายละเอียดดังต่อไปนี้

1. ทดสอบทั้งสิ้น 50 ลูป, ทุก sites จะใส่งาน, ทดสอบทุก site โดยให้ golden device วิ่งจนครบ 4 sites
2. ทดสอบทั้งสิ้น 50 ลูป, ทุก sites จะใส่งาน, ทดสอบเฉพาะ site ที่มี golden device โดยให้ golden device วิ่งจนครบ 4 sites
3. ทดสอบทั้งสิ้น 50 ลูป, มีงานแค่ site เดียวคือ golden device, ทดสอบทุก site โดยให้ golden device วิ่งจนครบ 4 sites

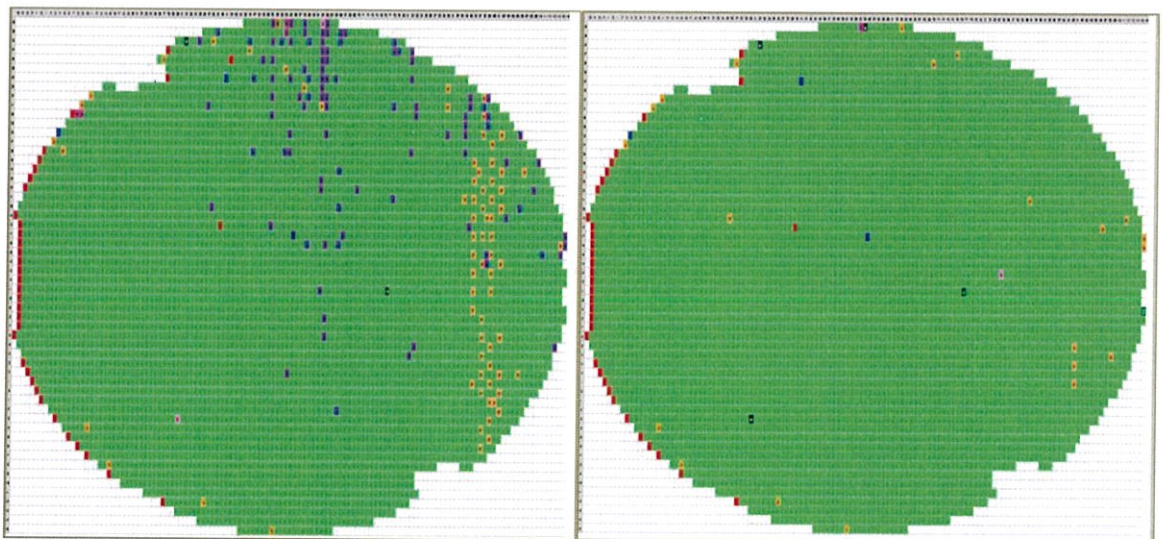
4. ทดสอบทั้งสิ้น 50 ลูป, มีงานแค่ site เดียวคือ golden device, ทดสอบเฉพาะ site ที่มี golden device โดยให้ golden device วิ่งจนครบ 4 sites

ผลการทดสอบของ 4 เงื่อนไขจะพิจารณาออกเป็น 2 คู่คือ เงื่อนไขที่1 และ2ควรได้ค่าใกล้เคียงกัน และเงื่อนไขที่3 และ4 ควรได้ค่าใกล้เคียงกัน จากข้อมูลที่ได้พบว่า test ที่มีค่า R&R, Reproducibility และ Repeatability มีค่ามากกว่า 7% ของเงื่อนไขที่1 และ2 กับเงื่อนไขที่3 และ4 ไม่ได้เกิดจากตัวโปรแกรมแต่เป็นเพราะ instrument interface ซึ่งเป็นเรื่องของฮาร์ดแวร์

3.1.4.6 Pilot run

เป็นการทดสอบแผ่นเวเฟอร์จริงไม่ได้ใช้ตัวงานในการทดสอบ ทำเหมือนกับทดสอบจริงที่ไลน์การผลิตแต่ แต่ขั้นตอนนี้ทดสอบกับเวเฟอร์แผ่นเดียว เป็น Qualification activities อันสุดท้าย ทำทั้งที่ Old CPU และ New CPU แล้ววิเคราะห์ค่าทางสถิติ CpkN ควรค่ามากกว่า 1.67 และคาดหวังว่าโปรแกรมใหม่นั้นไม่ทำให้เกิด Low Yield

ผลการทดสอบด้วย Pilot run บน wafer map



เวเฟอร์ที่ทดสอบด้วย Old CPU TPGM

เวเฟอร์ที่ทดสอบด้วย New CPU TPGM

รูปที่ 3.10 แสดงเวเฟอร์ที่ทดสอบด้วย Old CPU และ New CPU TPGM

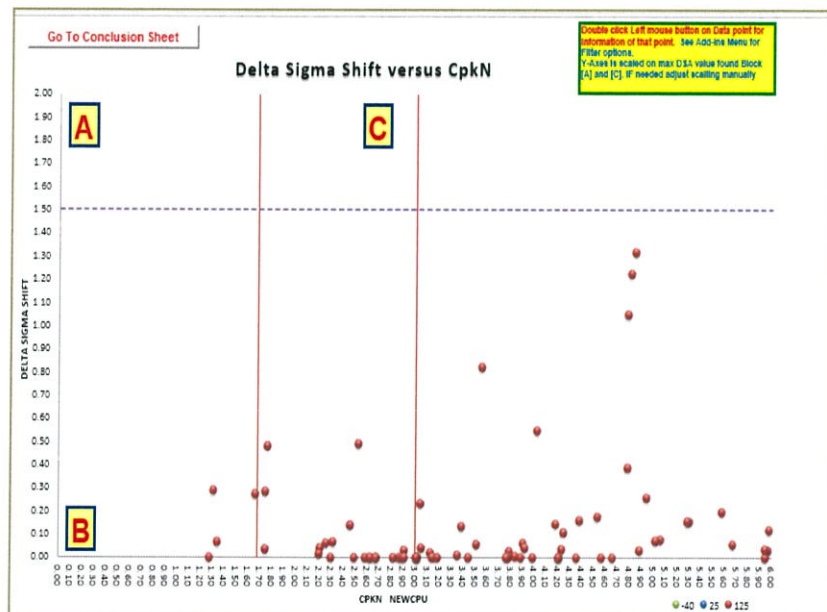
จากภาพจะเห็นได้ว่า wafer map ของเวเฟอร์ที่ทดสอบด้วย Old CPU TPGM นั้นให้ค่า CpkN ต่ำกว่าของ New CPU TPGM ซึ่งมันควรจะใกล้เคียงกันแต่การเพี้ยนที่เกิดขึ้นเป็นเรื่องของ contact sensitive invalid fail ไม่ใช่ปัญหาจากตัวโปรแกรม

DSA Analysis (Delta Sigma Analysis)

เราจะพิจารณา 3 พื้นที่ในกราฟ DSA Analysis ซึ่งจะพิจารณา ดังนี้

- Block A : มีค่า CpkN < 1.67, เกิดการ shift > 1.5
- Block B : มีค่า CpkN < 1.67, เกิดการ shift < 1.5
- Block C : มีค่า CpkN > 1.67, เกิดการ shift > 1.5

จากผลที่ได้ในกราฟ DSA Analysis พบว่ามีทั้งสิ้น 4 test อยู่ในพื้นที่ Block B ทั้งหมด และวิเคราะห์การกระจายตัวของข้อมูลแล้วระหว่าง Old และ New CPU TPGM ว่ามีความใกล้เคียงกัน สามารถดูได้จากภาพ ดังต่อไปนี้



รูปที่ 3.11 แสดงรูปกราฟ DSA Analysis

หลังจากที่เราตรวจสอบแล้วว่าข้อมูลที่อ่านได้ถูกต้องแล้ว ไม่มีอะไรผิดพลาดในโปรแกรม หลังจากนั้นก็นำโปรแกรมที่สร้างขึ้นใหม่นี้ไปใช้ทดสอบจริงใน wafer test

บทที่ 4

ผลการทดสอบ

จากการที่เรา นำโปรแกรมที่สร้างขึ้นใหม่โดยใช้กับหน่วยประมวลผลกลางแบบใหม่ หรือ New CPU ไปใช้ทดสอบจริงใน wafer test ที่ไลน์การผลิต พบว่าสามารถลดเวลาการทดสอบ คุณสมบัติแผ่นเวเฟอร์ได้เมื่อเทียบกับโปรแกรมเก่าที่ใช้กับหน่วยประมวลผลกลางแบบเก่าหรือ Old CPU โดยที่ New CPU Test program ใช้เวลาทดสอบ 2.26 วินาทีต่อ 1 touchdown ในขณะที่ Old CPU Test program ใช้เวลาทดสอบ 2.9 วินาทีต่อ 1 touchdown สามารถลดเวลาทดสอบคิดเป็น 22% ส่งผลให้เพิ่มกำลังการผลิตหรือ wafer test capacity มากขึ้นด้วยโดยมีการเพิ่มขึ้นโดยพิจารณาจากข้อมูล ดังต่อไปนี้ ที่ New CPU Test program ใช้เวลาทดสอบ 1.35 ชั่วโมงต่อแผ่นเวเฟอร์ 1 แผ่น แต่ Old CPU Test program ใช้เวลาทดสอบ 1.69 ชั่วโมงต่อแผ่นเวเฟอร์ 1 แผ่น ลดเวลาไปครึ่งชั่วโมงต่อแผ่นเวเฟอร์ 1 แผ่น และยังสามารถพัฒนาจำนวน Yield ให้มี Yield ที่ New CPU Test program เพิ่มขึ้นด้วยเมื่อเปรียบเทียบกับ Old CPU Test program สามารถพัฒนาจำนวน Yield จาก 95.87% เป็น 98.8% นอกจากนี้แล้วการสร้างโปรแกรมใหม่ที่ใช้กับหน่วยประมวลผลกลางแบบใหม่หรือ New CPU นี้ยังทำกำไรให้กับบริษัทเพิ่มขึ้นจำนวนมากเลยทีเดียว

บทที่ 5

สรุปผลการทดสอบและข้อเสนอแนะ

จากการดำเนินงานโครงการตั้งแต่ขั้นตอนการเปลี่ยนฮาร์ดแวร์จาก Old CPU Board ให้เป็น New CPU Board (requirement of New CPU conversion) และทำการเปลี่ยน System configuration และ Hardware requirement เพื่อรองรับให้เป็นระบบ New CPU, ทำ risk analysis เพื่อ optimization New CPU code ให้ผ่านเป็น bin1, คาดเดาว่าจะมีความเสี่ยงเนื่องจากข้อมูลที่อ่านได้หลังการเปลี่ยนฮาร์ดแวร์ให้กับเครื่องทดสอบ (tester) เลยทำเป็น risk assessment ขึ้น และปัญหาของ risk assessment นั้นสามารถแก้ไขได้ด้วย Qualification activities ซึ่งประกอบไปด้วย Test program checker & Hot spot detector ซึ่งสามารถแก้ไขความเสี่ยงเกี่ยวกับ Hot switching ได้, การทำ Spike check สามารถแก้ไขความเสี่ยงเรื่อง Overshoot/undershoot ของแรงดันที่แต่ละ pin ของตัวงานได้, การทำ Stability check สามารถแก้ไขความเสี่ยงเรื่องของ Unstable testing ได้, การทำ MSC(Measurement System Comparision) และ Site dependency สามารถแก้ไขความเสี่ยงเกี่ยวกับ Measurement shift ได้ และการทำ pilot run สามารถแก้ไขความเสี่ยงเกี่ยวกับ Unjustified Yield Loss ได้ หลังจากที่เราทำ Qualification activities เสร็จแล้วมีการตรวจสอบข้อมูลที่อ่านได้แล้วว่ามีความถูกต้องตรงตามเกณฑ์ที่ตั้งไว้ ก็ทำการนำตัวโปรแกรมที่สร้างใหม่นั้นไปทดสอบแผ่นเวเฟอร์จริงที่ wafer test ในไลน์การผลิต

ดังนั้นสรุปได้ว่าการพัฒนาโปรแกรมแบบเก่าที่ใช้กับหน่วยประมวลผลกลางแบบเก่ามาเป็นโปรแกรมที่ใช้กับหน่วยประมวลผลกลางแบบใหม่นั้น สามารถนำไปใช้ทดสอบคุณสมบัติแผ่นเวเฟอร์ที่ไลน์การผลิตของ wafer test ได้จริง และสามารถทำงานได้อย่างมีประสิทธิภาพตรงตามวัตถุประสงค์

ข้อเสนอแนะ

- การวางแผนงานเป็นขั้นตอนอย่างมีประสิทธิภาพ จะทำให้ทำงานได้อย่างบรรลุเป้าหมายและทันเวลาที่คาดการณ์ไว้
- การเข้าใจตัวโปรแกรมว่ามีการทดสอบอย่างไร จะทำให้การแก้ปัญหาง่ายขึ้น เนื่องจากเราเข้าใจหลักการทำงานของการทำงานของการทดสอบในแต่ละส่วนแล้ว

เอกสารอ้างอิง

- [1] นาย Matio Marcus. การเขียนโปรแกรมภาษา C++. สืบค้นวันที่ 14 ธันวาคม 2560, เข้าถึงได้จาก : <http://marcuscode.com/lang/cpp>
- [2] หนังสือ C372MX Programming Training Basic Programming Training Workbook. สืบค้นวันที่ 16 ธันวาคม 2560