บทที่ 6

ผลและการวิเคราะห์การทวนสอบด้วยวิธีจำลองการทำงาน

แต่ละโมดูลอธิบายการทำงานในระดับของอาร์ทีแอล เนื่องจากถ้าเพิ่มรายละเอียดของการทำงานจะทำให้เป็นการเพิ่มความซับซ้อนในโมดูลที่จะทดสอบ ทำให้เกิดปัญหาในการทวนสอบและการแทนโมดูลด้วยวิธีอย่างมีแบบแผน จึงทำให้แต่ละโมดูลมีความเป็นหลักนามธรรมมากขึ้น โดยในส่วนรายละเอียดการทำงานได้มีการทดสอบด้วยวิธีจำลองการทำงานไว้แล้วในขั้นตอนของการออกแบบดังหัวข้อที่ 5.1 นอกจากนี้ได้นำวิธีการทดสอบมาประยุกต์ใช้ทั้งในการทวนสอบอย่างมีแบบแผนด้วยวิธีการของ black-box testing ในหัวข้อ 4.4 และการทวนสอบด้วยการจำลองการทำงานได้นำหลักของการทดสอบแบบ white box-testing และ interface testing มาใช้ ทั้งนี้ในงานวิจัยไม่ได้รวมถึงการทวนสอบอย่างมีแบบแผน จึงนำผลที่ได้จากการเสนอและการทดลองเบื้องต้นอยู่ในภาคผนวก ข. ส่วนในงานวิจัยหลักจะเป็นการทวนสอบด้วยวิธีการจำลองการทำงาน

การทดสอบจะใช้ชุดทดสอบที่เลือกโดยการแบ่งกลุ่มตามหัวข้อที่ 6.1 และทำการทดสอบโดยการตรวจผลลัพธ์ และอื่นๆดังแสดงในตารางที่ 6.1-6.5 ชุดคำสั่งทดสอบทั้งหมดจะถูกนำเอาเข้าสู่ไมโครโพรเซสเซอร์อย่างต่อเนื่อง และคำนึงถึงการทดสอบในไปป์ไลน์เช่นการทดสอบปัญหาจากข้อมูลและการควบคุมในไปป์ไลน์ด้วย โดยผลลัพธ์ที่ถูกต้องได้จากการนำชุดทดสอบชุดเดิมเข้าทำงานในส่วนของโปรแกรมหาจุดบกพร่องและทำการเปรียบเทียบกับผลที่ได้จากการจำลองการทำงาน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

6.1 ผลการวิเคราะห์ชุดทดสอบไมโครโพรเซสเซอร์อาร์ม 7

ตารางที่6.1 ผลการทดสอบด้วยชุดคำสั่งในกรณีต่างๆ(1)

Instruction	Function	OPCODE	PSR (Flag)				Access	Result	B /	Number	Addressing Mode			
			N	V	Z	C	Register		W	Cycle	Direct	Immed.	Index	Base Index
AND (S=0)	Op1/Register AND Op2/Register	E0SXX00X	/	/	/	/	/	/	/	1, 3	/			
ANDS (S=1)	Op1/Register AND Op2/Immediate value	E2SXX0XX	/	/	/	/	/	/	/	1, 3	/	/		
V = Even	Op1/Register AND Op2/Register with Shift immediate value	E0SXXXVX	/	/	/	/	/	/	/	2, 4	/	/		
O = Odd	Op1/Register AND Op2/Register with Shift Register(Rs)	E0SXXXOX	/	/	/	/	/	/	/	2, 4	/			
EOR (S=2)	Op1/Register AND Op2/Register	E0SXX00X	/	/	/	/	/	/	/	1, 3	/			
EORS (S=3)	Op1/Register AND Op2/Immediate value	E2SXX0XX	/	/	/	/	/	/	/	1, 3	/	/		
V = Even	Op1/Register AND Op2/Register with Shift immediate value	E0SXXXVX	/	/	/	/	/	/	/	2, 4	/	/		
O = Odd	Op1/Register AND Op2/Register with Shift Register(Rs)	E0SXXXOX	/	/	/	/	/	/	/	2, 4	/			
SUB (S=4)	Op1/Register AND Op2/Register	E0SXX00X	/	/	/	/	/	/	/	1, 3	/			
SUBS (S=5)	Op1/Register AND Op2/Immediate value	E2SXX0XX	/	/	/	/	/	/	/	1, 3	/	/		
V = Even	Op1/Register AND Op2/Register with Shift immediate value	E0SXXXVX	/	/	/	/	/	/	/	2, 4	/	/		
O = Odd	Op1/Register AND Op2/Register with Shift Register(Rs)	E0SXXXOX	/	/	/	/	/	/	/	2, 4	/			
RSB (S=6)	Op1/Register AND Op2/Register	E0SXX00X	/	/	/	/	/	/	/	1, 3	/			
RSBS (S=7)	Op1/Register AND Op2/Immediate value	E2SXX0XX	/	/	/	/	/	/	/	1, 3	/	/		
V = Even	Op1/Register AND Op2/Register with Shift immediate value	E0SXXXVX	/	/	/	/	/	/	/	2, 4	/	/		
O = Odd	Op1/Register AND Op2/Register with Shift Register(Rs)	E0SXXXOX	/	/	/	/	/	/	/	2, 4	/			

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่6. 2 ผลการทดสอบด้วยชุดคำสั่งในกรณีต่างๆ (2)

Instruction	Function	OPCODE	PSR (Flag)				Access	Result	B /	Number	Addressing Mode			
			N	V	Z	C	Register		W	Cycle	Direct	Immed.	Index	Base Index
ADD (S=8)	Op1/Register AND Op2/Register	E0SXX00X	/	/	/	/	/	/	/	1, 3	/			
ADDS (S=9)	Op1/Register AND Op2/Immediate value	E2SXX0XX	/	/	/	/	/	/	/	1, 3	/	/		
V = Even	Op1/Register AND Op2/Register with Shift immediate value	E0SXXXVX	/	/	/	/	/	/	/	2, 4	/	/		
O = Odd	Op1/Register AND Op2/Register with Shift Register(Rs)	E0SXXXOX	/	/	/	/	/	/	/	2, 4	/			
ADC (S=A)	Op1/Register AND Op2/Register	E0SXX00X	/	/	/	/	/	/	/	1, 3	/			
ADCS (S=B)	Op1/Register AND Op2/Immediate value	E2SXX0XX	/	/	/	/	/	/	/	1, 3	/	/		
V = Even	Op1/Register AND Op2/Register with Shift immediate value	E0SXXXVX	/	/	/	/	/	/	/	2, 4	/	/		
O = Odd	Op1/Register AND Op2/Register with Shift Register(Rs)	E0SXXXOX	/	/	/	/	/	/	/	2, 4	/			
SBC (S=C)	Op1/Register AND Op2/Register	E0SXX00X	/	/	/	/	/	/	/	1, 3	/			
SBCS (S=D)	Op1/Register AND Op2/Immediate value	E2SXX0XX	/	/	/	/	/	/	/	1, 3	/	/		
V = Even	Op1/Register AND Op2/Register with Shift immediate value	E0SXXXVX	/	/	/	/	/	/	/	2, 4	/	/		
O = Odd	Op1/Register AND Op2/Register with Shift Register(Rs)	E0SXXXOX	/	/	/	/	/	/	/	2, 4	/			
RSC (S=E)	Op1/Register AND Op2/Register	E0SXX00X	/	/	/	/	/	/	/	1, 3	/			
RSCS (S=F)	Op1/Register AND Op2/Immediate value	E2SXX0XX	/	/	/	/	/	/	/	1, 3	/	/		
V = Even	Op1/Register AND Op2/Register with Shift immediate value	E0SXXXVX	/	/	/	/	/	/	/	2, 4	/	/		
O = Odd	Op1/Register AND Op2/Register with Shift Register(Rs)	E0SXXXOX	/	/	/	/	/	/	/	2, 4	/			

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่6.3 ผลการทดสอบด้วยชุดคำสั่งในกรณีต่างๆ (3)

Instruction	Function	OPCODE	PSR (Flag)				Access	Result	B /	Number	Addressing Mode			
			N	V	Z	C	Register		W	Cycle	Direct	Immed.	Index	Base Index
TST (S=1)	Op1/Register AND Op2/Register	E1SXX00X	/	/	/	/	/	/	/	1, 3	/			
V = Even	Op1/Register AND Op2/Immediate value	E3SXX00X	/	/	/	/	/	/	/	1, 3	/	/		
O = Odd	Op1/Register AND Op2/Register with Shift immediate value	E1SXXXVX	/	/	/	/	/	/	/	2, 4	/	/		
	Op1/Register AND Op2/Register with Shift Register(Rs)	E1SXXXOX	/	/	/	/	/	/	/	2, 4	/			
TEQ (S=3)	Op1/Register AND Op2/Register	E1SXX00X	/	/	/	/	/	/	/	1, 3	/			
V = Even	Op1/Register AND Op2/Immediate value	E3SXX00X	/	/	/	/	/	/	/	1, 3	/	/		
O = Odd	Op1/Register AND Op2/Register with Shift immediate value	E1SXXXVX	/	/	/	/	/	/	/	2, 4	/	/		
	Op1/Register AND Op2/Register with Shift Register(Rs)	E1SXXXOX	/	/	/	/	/	/	/	2, 4	/			
CMP (S=5)	Op1/Register AND Op2/Register	E1SXX00X	/	/	/	/	/	/	/	1, 3	/			
V = Even	Op1/Register AND Op2/Immediate value	E3SXX00X	/	/	/	/	/	/	/	1, 3	/	/		
O = Odd	Op1/Register AND Op2/Register with Shift immediate value	E1SXXXVX	/	/	/	/	/	/	/	2, 4	/	/		
	Op1/Register AND Op2/Register with Shift Register(Rs)	E1SXXXOX	/	/	/	/	/	/	/	2, 4	/			
CMN (S=7)	Op1/Register AND Op2/Register	E1SXX00X	/	/	/	/	/	/	/	1, 3	/			
V = Even	Op1/Register AND Op2/Immediate value	E3SXX00X	/	/	/	/	/	/	/	1, 3	/	/		
O = Odd	Op1/Register AND Op2/Register with Shift immediate value	E1SXXXVX	/	/	/	/	/	/	/	2, 4	/	/		
	Op1/Register AND Op2/Register with Shift Register(Rs)	E1SXXXOX	/	/	/	/	/	/	/	2, 4	/			

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่6. 4 ผลการทดสอบด้วยชุดคำสั่งในกรณีต่างๆ (4)

Instruction	Function	OPCODE	PSR (Flag)				Access	Result	B /	Number	Addressing Mode			
			N	V	Z	C	Register		W	Cycle	Direct	Immed.	Index	Base Index
ORR (S=8)	Op1/Register AND Op2/Register	E1SXX00X	/	/	/	/	/	/	/	1, 3	/			
ORRS (S=9)	Op1/Register AND Op2/Immediate value	E3SXX00X	/	/	/	/	/	/	/	1, 3	/	/		
V = Even	Op1/Register AND Op2/Register with Shift immediate value	E1SXXXVX	/	/	/	/	/	/	/	2, 4	/	/		
O = Odd	Op1/Register AND Op2/Register with Shift Register(Rs)	E1SXXXOX	/	/	/	/	/	/	/	2, 4	/			
MOV (S=A)	Op1/Register AND Op2/Register	E1SXX00X	/	/	/	/	/	/	/	1, 3	/			
MOVS (S=B)	Op1/Register AND Op2/Immediate value	E3SXX00X	/	/	/	/	/	/	/	1, 3	/	/		
V = Even	Op1/Register AND Op2/Register with Shift immediate value	E1SXXXVX	/	/	/	/	/	/	/	2, 4	/	/		
O = Odd	Op1/Register AND Op2/Register with Shift Register(Rs)	E1SXXXOX	/	/	/	/	/	/	/	2, 4	/			
BIC (S=C)	Op1/Register AND Op2/Register	E1SXX00X	/	/	/	/	/	/	/	1, 3	/			
BICS (S=D)	Op1/Register AND Op2/Immediate value	E3SXX00X	/	/	/	/	/	/	/	1, 3	/	/		
V = Even	Op1/Register AND Op2/Register with Shift immediate value	E1SXXXVX	/	/	/	/	/	/	/	2, 4	/	/		
O = Odd	Op1/Register AND Op2/Register with Shift Register(Rs)	E1SXXXOX	/	/	/	/	/	/	/	2, 4	/			
MVN (S=E)	Op1/Register AND Op2/Register	E1SXX00X	/	/	/	/	/	/	/	1, 3	/			
MVN (S=F)	Op1/Register AND Op2/Immediate value	E3SXX00X	/	/	/	/	/	/	/	1, 3	/	/		
V = Even	Op1/Register AND Op2/Register with Shift immediate value	E1SXXXVX	/	/	/	/	/	/	/	2, 4	/	/		
O = Odd	Op1/Register AND Op2/Register with Shift Register(Rs)	E1SXXXOX	/	/	/	/	/	/	/	2, 4	/			

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่6. 5 ผลการทดสอบด้วยชุดคำสั่งในกรณีต่างๆ (5)

Instruction	Function	OPCODE	PSR (Flag)				Access Register	Result	B / W	Number Cycle	Addressing Mode			
			N	V	Z	C					Direct	Immed.	Index	Base Index
B	Branch not have Condition	XAXXXXXX	-	-	-	-	/	/	/	3			/	
BL	Branch,with Link (could return)	XBXXXXXX	-	-	-	-	/	/	/	3			/	
MUL	Multiply (Rd = Rs*Rm)	E0SX0X9X	/	/	/	/	/	/	/	>2	/		/	/
MLA	Multiply (Rd = Rs*Rm + Rn)	E0SXXX9X	/	/	/	/	/	/	/	>2	/		/	/
LDR	Add/Sub Offset Before/After From Register/Immediate Value with/without Shift and Write/not write Back		-	-	-	-	/	/	/	3, 5	/		/	/
STR	Add/Sub Offset Before/After From Register/Immediate Value with/without Shift and Write/not write Back		-	-	-	-	/	/	/	2	/		/	/
LDM	Add/Sub Offset Before/After with Write/not write Back		-	-	-	-	/	/	/	3, 5, >5	/		/	/
STM	Add/Sub Offset Before/After with Write/not write Back		-	-	-	-	/	/	/	2, >2	/		/	/
SWP	Swap Data between Register and Memory	E10XX09X	-	-	-	-	/	/	/	4	/		/	/

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

การวิเคราะห์ชุดทดสอบในกลุ่มของ Data Processing

จากตาราง 6.1 – 6.5 เป็นการสรุปชุดทดสอบที่สร้างขึ้นโดยแสดงให้เห็นว่าในแต่ละคำสั่งสามารถทดสอบส่วนใดได้บ้าง ตารางที่ 6.1 – 6.4 เป็นส่วนของชุดทดสอบคำสั่งในกลุ่มของ Data Processing โดยคำสั่งในกลุ่มนี้จะขึ้นจะมีค่ารหัสดำเนินการ (Op code) โดยให้ไบต์บนสุดเป็นไบต์ที่ 8 และล่างสุดเป็นไบต์ที่ 1 ตาม รายละเอียดดังนี้

ไบต์ที่ 8 : จะมีค่าเป็น E เสมอเนื่องจากเป็นส่วนของบิตเงื่อนไขถ้าต้องการให้มีการทำคำสั่งนั้นจะต้องให้ค่าเป็น E

ไบต์ที่ 7 : สามารถแบ่งออกเป็น 2 กลุ่มได้คือ

- กลุ่มของคำสั่งทางคณิตศาสตร์จะมีค่าเป็น 0 หรือ 2 เท่านั้นโดยที่จะเป็นค่า 2 ก็ต่อเมื่อมีการเรียกใช้ตัวถูกดำเนินการเป็นค่าจำนวนเต็มเท่านั้น
- กลุ่มของคำสั่งทางลอจิก จะมีค่าเป็น 1 หรือ 3 เท่านั้น โดยที่จะมีค่าเป็น 3 ก็ต่อเมื่อมีการเรียกใช้ตัวถูกดำเนินการเป็นค่าจำนวนเต็มเท่านั้น

ไบต์ที่ 6 : เป็นค่าบอกถึงการคิดค่าตัวบ่งชี้หรือไม่ เนื่องจากไมโครโพรเซสเซอร์อาร์ม 7 สามารถให้ทำงานทั้งแบบที่คิดและไม่คิดค่าตัวบ่งชี้ ในที่นี้แทนด้วยตัวแปร S

ไบต์ที่ 4,5 : เป็นค่าใดก็ได้ทั้งนี้คือส่วนของการอ้างถึงรีจิสเตอร์หรือเป็นค่าจำนวนเต็ม

ไบต์ที่ 3 : เป็นการบอกว่ามีการเลื่อนค่าในคำสั่งทางคณิตศาสตร์และลอจิกด้วยหรือไม่ โดยที่ ค่า 0 หมายถึงไม่มีการเลื่อนค่า นอกจากนี้จะมีการเลื่อนค่าซึ่งจะเป็นจำนวนที่ต้องการเลื่อน

ไบต์ที่ 2 เป็นตัวบอกความแตกต่างของรูปแบบของแต่ละฟังก์ชั่นการทำงาน โดยแบ่งเป็น 2 กลุ่มคือ

- กลุ่มที่ไม่มีการเลื่อนค่า (ดูจากไบต์ที่ 3) ถ้าค่าในไบต์ที่ 2 เป็น 0 หมายถึงการใช้รีจิสเตอร์เป็นทั้งตัวดำเนินการและตัวถูกดำเนินการ นอกจากนั้นเป็นการใช้ตัวถูกดำเนินการเป็นจำนวนเต็ม
- กลุ่มที่มีการเลื่อนค่า (ดูจากไบต์ที่ 3) ถ้าไบต์นี้มีค่าเป็นเลขคู่ (V = Even) หมายถึงการเลื่อนค่า โดยมีจำนวนครั้งจากจำนวนเต็ม แต่ถ้าเป็นเลขคี่ (O = Odd) หมายถึงการเลื่อนค่าโดยใช้จำนวนครั้งในการเลื่อนจากรีจิสเตอร์ไบต์ล่าง

ไบต์ที่ 1 : เป็นค่าใดก็ได้

จากข้างต้นเป็นรูปแบบของการทำงานในกลุ่มของ Data Processing ที่สามารถเกิดขึ้นได้ทั้งหมดโดยการออกแบบส่วนของการถอดรหัสจะใช้ลักษณะเช่นนี้ในการแบ่งกลุ่มโดยจะเริ่มตรวจค่าจากไบต์บนลงสู่ไบต์ล่าง ในตารางที่ 6.1- 6.4 จะแสดงส่วนต่างที่ได้ครอบคลุมการทดสอบซึ่งได้จากการสร้างชุดทดสอบตามลักษณะการแบ่งค่ารหัสดำเนินการเช่นนี้ ค่าที่สามารถทดสอบได้แก่ ค่าตัวบ่งชี้ การทดสอบในส่วนของการเรียกใช้ รีจิสเตอร์ การตรวจสอบผลของการทำงานในส่วนของหน่วยคำนวณและ

ตรรกะ หน่วยการเลื่อนค่า การรับและส่งข้อมูลออกสู่บัสของข้อมูล การทำงานในลักษณะของไบต์และเวิร์ด ตรวจสอบจำนวนของคาบที่ใช้ในการทำงาน และการอ้างอิงเลขที่อยู่โดยสามารถอ้างได้ 2 แบบคือ ส่วนของ Direct และ immediate เราสามารถสรุปส่วนต่างๆและสิ่งที่ได้ทำการทวนสอบโดยการใช้ชุดทดสอบของกลุ่ม Data Processing ดังตารางที่ 6.6.

ตารางที่ 6.6 ส่วนต่างๆที่ได้ทำการทวนสอบด้วยชุดคำสั่ง Data Processing

หน่วยที่มีการเรียกใช้	สิ่งที่ได้ทำการทวนสอบ	หมายเหตุ
หน่วยรีจิสเตอร์	การอ่าน / เขียน	ทุกคำสั่งที่มีการใช้ตัวถูกดำเนินการจากรีจิสเตอร์และเก็บผลลัพธ์สู่รีจิสเตอร์
	สัญญาณอ่าน /เขียน	
	การเรียกใช้รีจิสเตอร์	มีการเรียกใช้รีจิสเตอร์ทุกตัวในชุดทดสอบกลุ่มของ Data Processing
บัสข้อมูลขนาด 32 บิต	การอ่านค่าจากบัส	
	การเขียนค่าลงสู่บัส	
หน่วยของการคูณ	การผ่านค่า	เนื่องจากไม่มีคำสั่งการคูณ
หน่วยการเลื่อนค่า	การเลื่อนค่า ทั้ง 4 แบบ	
	การผ่านค่า	ในกรณีที่ทำคำสั่งที่ไม่มีการเลื่อนค่า
หน่วยควบคุม	สัญญาณที่ควบคุมแต่ละหน่วย	
	ข้อมูลที่เป็นจำนวนเต็ม	ในกรณีที่มีการเรียกใช้จำนวนเต็ม
	การทำงานของไปป์ไลน์	
หน่วยคำนวณและตรรกะ	การประมวลผล	
ปัญหาของไปป์ไลน์	ปัญหาจากข้อมูล	โดยชุดคำสั่งที่ทดสอบจะมีการเรียกใช้รีจิสเตอร์ผลลัพธ์ของคำสั่งก่อนหน้ามาเป็นตัวถูกดำเนินการในคำสั่งถัดไป
ค่าอื่นๆ	ค่าตัวบ่งชี้	ในกรณีที่ใช้ชุดคำสั่งที่มีการเปลี่ยนค่าตัวบ่งชี้
	การเรียกใช้ไบต์ และเวิร์ด	

การวิเคราะห์ชุดทดสอบในกลุ่มของการกระโดดข้ามการทำงาน

ในส่วนของการกระโดดข้ามการทำงานจะทำการแบ่งออกเป็น 2 ชุดหลักคือ การกระโดดข้ามการทำงานแบบเก็บค่าเลขชี้ตำแหน่งเดิม และการกระโดดข้ามแบบไม่มีเงื่อนไข โดยมีความแตกต่างของรหัสดำเนินการ ณ. ไบต์ที่ 7 คือจะมีค่า B และ A ในเลขฐานสิบหกตามลำดับ การทำงานในส่วนนี้จะไม่มีผลต่อ

ค่าตัวบ่งชี้ จึงมีการตรวจสอบเฉพาะส่วนของการเรียกใช้รีจิสเตอร์ ผลที่ได้หมายถึงค่าตำแหน่งของเลขที่อยู่ที่เปลี่ยนไปซึ่งรวมถึงในกรณีที่มีการเก็บค่าลงสู่รีจิสเตอร์ที่ 14 และ 15 นอกจากนี้มีการตรวจผลของการใช้คาบเวลาในการทำงาน รวมถึงการอ้างอิงเลขที่อยู่ซึ่งจะมีเฉพาะแบบเลขที่อยู่เชิงดรรชนี (Index addressing) การใช้ชุดทดสอบในกลุ่มการกระโดดข้ามการทำงานจะมีผลต่อการทำงานของไปป์ไลน์ ดังนั้นการทดสอบโดยป้อนชุดทดสอบอย่างต่อเนื่องจะช่วยเป็นการตรวจสอบการทำงานของไปป์ไลน์และการเกิดปัญหาชนิดควบคุมของไปป์ไลน์ได้อีกด้วย ชุดทดสอบสามารถครอบคลุมได้ดังตารางที่ 6.7

ตารางที่ 6.7 ส่วนต่างๆที่ได้ทำการทวนสอบด้วยชุดคำสั่งกระโดดข้ามการทำงาน

หน่วยที่มีการเรียกใช้	สิ่งที่ได้ทำการทวนสอบ	หมายเหตุ
หน่วยควบคุม	สัญญาณที่ควบคุมแต่ละหน่วย	
	ข้อมูลที่เป็นจำนวนเต็ม	ในกรณีที่มีการเรียกใช้จำนวนเต็ม
	การทำงานของไปป์ไลน์	
	ค่าเลขชี้ตำแหน่ง	เนื่องจากคำสั่งนี้มีผลกับค่าเลขชี้ตำแหน่งแต่ไม่มีผลต่อการทำงานในส่วนอื่น
ปัญหาของไปป์ไลน์	ปัญหาจากการควบคุม	โดยใช้วิธีการขยายคาบการทำงานและหยุดการทำงานอื่นๆจนกว่าจะมีการเอาคำสั่งใหม่เข้ามา
ค่าอื่นๆ	ค่าตัวบ่งชี้	ในกรณีที่ใช้ชุดคำสั่งที่มีการเปลี่ยนค่าตัวบ่งชี้

การวิเคราะห์ชุดทดสอบในกลุ่มของการคูณ

การคูณจะมี 2 แบบคือการคูณแบบธรรมดาและการคูณที่มีการบวกค่ารวมด้วย ซึ่งความแตกต่างจะอยู่ที่ไบต์ที่ 4 ของรหัสดำเนินการ โดยถ้ามีค่า 0 หมายถึงการคูณแบบธรรมดา นอกจากนั้นจะเป็นการคูณแบบบวกค่าด้วย การคูณทั้งสองแบบจะมีผลต่อค่าตัวบ่งชี้จึงมีการตรวจสอบในส่วนนี้ด้วย รวมถึงค่าผลลัพธ์ที่ได้ การเรียกใช้รีจิสเตอร์ การใช้งานแบบไบต์และเวิร์ด โดยมีการอ้างอิงเลขที่อยู่แบบการเข้าถึงโดยตรง การอ้างอิงแบบเลขที่อยู่เชิงดรรชนี และแบบดรรชนีเลขฐาน (Base Index) การทำงานของการคูณนี้จะใช้เวลาในการทำงานมากกว่า 2 คาบทั้งนี้ขึ้นอยู่กับจำนวนของเลขที่ต้องการคูณ

ตารางที่ 6.8 ส่วนต่างๆที่ได้ทำการทวนสอบด้วยชุดคำสั่งคูณ

หน่วยที่มีการเรียกใช้	สิ่งที่ได้ทำการทวนสอบ	หมายเหตุ
หน่วยรีจิสเตอร์	การอ่าน / เขียน	เมื่อมีการอ่านและเขียนข้อมูลจากรีจิสเตอร์เพื่อมาทำการคูณ
	สัญญาณอ่าน /เขียน	
	การเรียกใช้รีจิสเตอร์	เมื่อต้องการเก็บและเรียกตัวถูกดำเนินการจากรีจิสเตอร์
บัสข้อมูลขนาด 32 บิต	การอ่านค่าจากบัส	
	การเขียนค่าลงสู่บัส	
หน่วยของการคูณ	การคูณค่า	ทั้ง 2 แบบ คือ MUL และ MLA
หน่วยควบคุม	สัญญาณที่ควบคุมแต่ละหน่วย	
	ข้อมูลที่เป็นจำนวนเต็ม	ในกรณีที่มีการเรียกใช้จำนวนเต็ม
	การทำงานของไปป์ไลน์	
หน่วยคำนวณและตรรกะ	การประมวลผล	
ปัญหาของไปป์ไลน์	ปัญหาจากข้อมูล	โดยชุดคำสั่งที่ทดสอบจะมีการเรียกใช้รีจิสเตอร์ผลลัพธ์ของคำสั่งก่อนหน้ามาเป็นตัวถูกดำเนินการในคำสั่งถัดไป.
ค่าอื่นๆ	ตัวบ่งชี้	ในกรณีที่ใช้ชุดคำสั่งที่มีการเปลี่ยนค่าตัวบ่งชี้
	การใช้ไบต์ และเวิร์ด	

การวิเคราะห์ชุดทดสอบในกลุ่มของการติดต่อหน่วยความจำ

1. คำสั่งในกลุ่มแรกนี้ไม่สามารถจัดแบ่งได้อย่างแน่นอน โดยมีคำสั่งที่สามารถทำงานได้ 4 แบบ คือ การเก็บข้อมูลลงหน่วยความจำแบบเดี่ยว แบบหลายทาง การเอาข้อมูลจากหน่วยความจำแบบเดี่ยว แบบหลายทาง การทดสอบจะตรวจการเรียกใช้รีจิสเตอร์ ผลที่ได้จากการทำงาน จำนวนของคาบเวลาที่ใช้งาน รวมถึงการอ้างเลขที่อยู่ทั้งแบบการเข้าถึงโดยตรง การอ้างอิงแบบเลขที่อยู่เชิงดรรชนี และแบบดรรชนีเลขฐาน (Base Index) การใช้งานแบบไบต์และเวิร์ด การทำงานปกติในแต่ละคำสั่งจะมีการใช้คาบเวลาไม่เท่ากัน โดยคำสั่งของการเอาข้อมูลจากหน่วยความจำแบบหลายทางจะใช้ถึง 3, 5 หรือมากกว่านั้น (>5) ทั้งนี้ขึ้นกับว่าจะมีการเอาข้อมูลจากหน่วยความจำลงสู่รีจิสเตอร์กี่ตัว ทำนองเดียวกันนี้จึงทำให้การเก็บข้อมูลลงหน่วยความจำมีการใช้คาบเวลา มากกว่าหรือเท่ากับ 2 คาบ

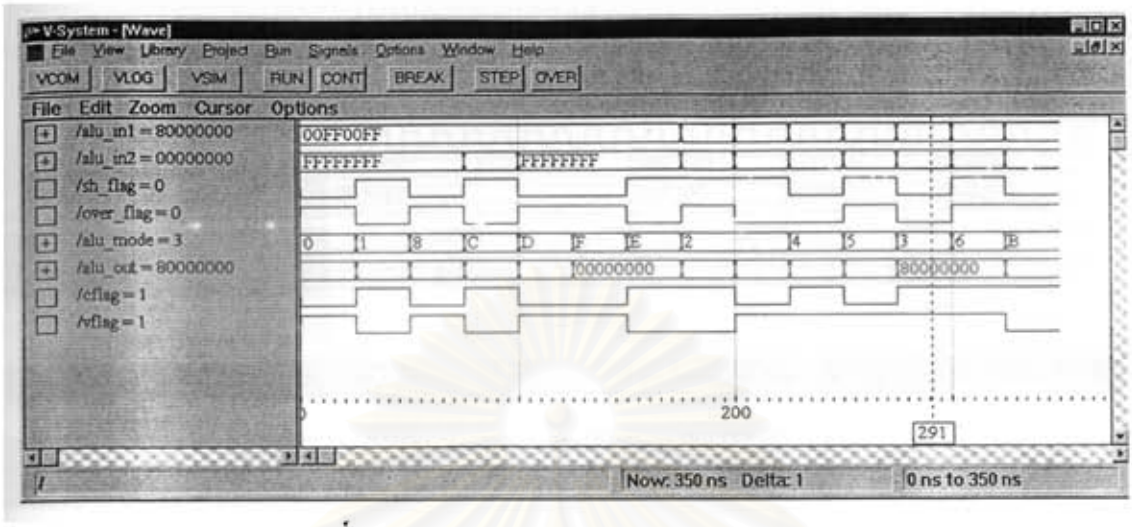
2. ในส่วนของการแลกเปลี่ยนข้อมูลระหว่างหน่วยความจำและรีจิสเตอร์ จะมีค่ารหัสดำเนินการแน่นอน และคาบเวลาใช้งานคงที่คือ 4 มีการตรวจสอบในส่วนของการเรียกใช้รีจิสเตอร์ การอ้างข้อมูลแบบไบต์และเวิร์ด รวมถึงการอ้างอิงเลขที่อยู่แบบการเข้าถึงโดยตรง การอ้างอิงแบบเลขที่อยู่เชิงดรรชนี และแบบดรรชนีเลขฐาน (Base Index)

ตารางที่ 6.9 ส่วนต่างๆที่ได้ทำการทวนสอบด้วยชุดคำสั่งการติดต่อหน่วยความจำ

หน่วยที่มีการเรียกใช้	สิ่งที่ได้ทำการทวนสอบ	หมายเหตุ
หน่วยรีจิสเตอร์	การอ่าน / เขียน	ทุกคำสั่งที่มีการใช้ตัวถูกดำเนินการจากรีจิสเตอร์และเก็บผลลัพธ์สู่รีจิสเตอร์
	สัญญาณอ่าน /เขียน	
	การเรียกใช้รีจิสเตอร์	เมื่อต้องการอ่านค่าหรือเก็บค่าลงสู่รีจิสเตอร์
บัสข้อมูลขนาด 32 บิต	การอ่านค่าจากบัส	
	การเขียนค่าลงสู่บัส	
หน่วยของการคูณ	การผ่านค่า	เนื่องจากไม่มีคำสั่งการคูณ
หน่วยการเลื่อนค่า	การผ่านค่า	ในกรณีที่ทำคำสั่งที่ไม่มีการเลื่อนค่า
หน่วยควบคุม	สัญญาณที่ควบคุมแต่ละหน่วย	
	ข้อมูลที่เป็นจำนวนเต็ม	ในกรณีที่มีการเรียกใช้จำนวนเต็ม
	การทำงานของไปป์ไลน์	
	ค่าเลขชี้ตำแหน่ง	ในกรณีที่มีการเก็บค่าลงสู่รีจิสเตอร์ 15 ซึ่งจะเป็นรีจิสเตอร์เดียวกับพีซี
หน่วยคำนวณและตรรกะ	การประมวลผล	ในกรณีที่มีการคำนวณเลขที่ตำแหน่งในหน่วยความจำ
ปัญหาของไปป์ไลน์	ปัญหาจากข้อมูล	โดยชุดคำสั่งที่ทดสอบจะมีการเรียกใช้รีจิสเตอร์ผลลัพธ์ของคำสั่งก่อนหน้ามาเป็นตัวถูกดำเนินการในคำสั่งถัดไป
	ปัญหาจากการควบคุม	ในกรณีที่เก็บค่าลงสู่รีจิสเตอร์ที่ 15
ค่าอื่นๆ	ค่าตัวบ่งชี้	ในกรณีที่ใช้ชุดคำสั่งที่มีการเปลี่ยนค่าตัวบ่งชี้
	การเรียกใช้ไบต์ และเวิร์ด	

6.2 ผลจากการจำลองการทำงานด้วยชุดทดสอบที่กำหนด ตัวอย่างผลการทดลองของหน่วยต่างๆ

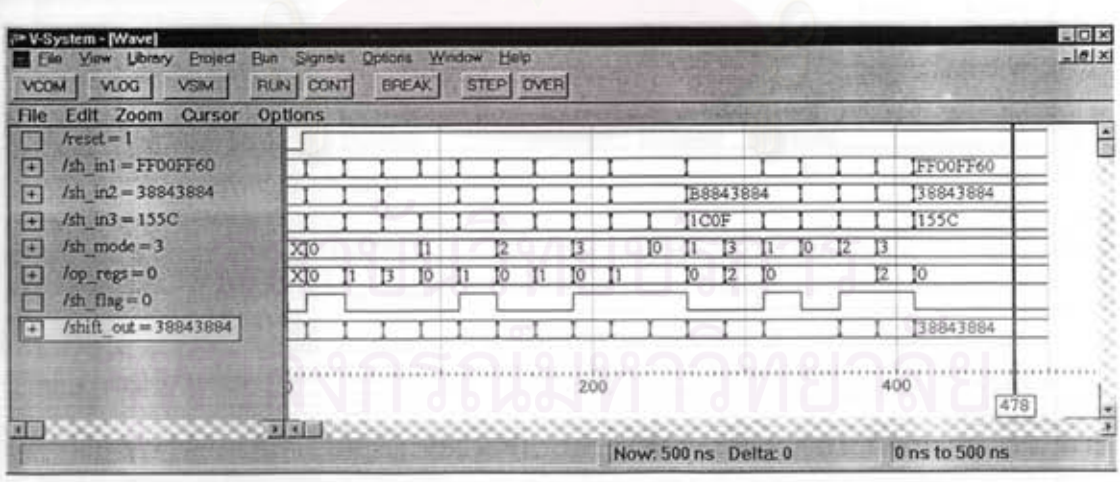
1. หน่วยคำนวณและตรรกะ



รูปที่6. 1 Waveform ของหน่วยคำนวณและตรรกะ

จากรูปที่ 6.1 เป็นผลของการทำทวนสอบการลบค่าตามคำสั่ง SUB r1,r2,r3 โดยที่ค่าใน r2 คือค่าใน alu_in1 r2 คือค่า alu_in2 และผลลัพธ์เก็บใน r3 ในรูปคือค่า alu_out ในที่นี้คือการบวกค่า 80000000 กับค่า 0 ผลที่ได้คือค่า 80000000 โดยไม่คำนึงถึงค่าตัวบ่งชี้จึงไม่ได้นำมาแสดงในรูป

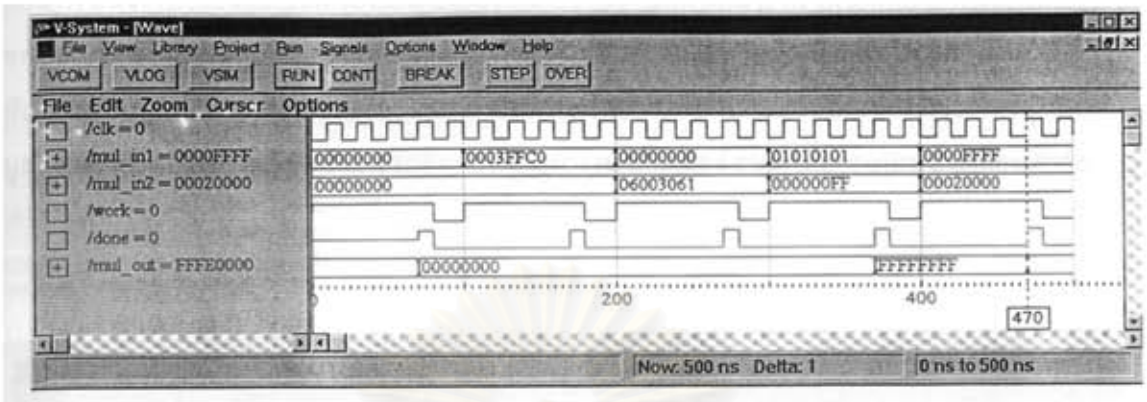
2. หน่วยการเลื่อนค่า



รูปที่6. 2 waveform ของหน่วยการเลื่อนค่า

จากรูปที่ 6.2 เป็นการแสดงผลของการผ่านค่าในหน่วยเลื่อนค่า โดยที่ผ่านเข้าไปทาง sh_in2 จะต้องออกสู่ผลลัพธ์จะเหมือนเดิม ดังในค่าของ shift_out ตามรูป

3. หน่วยของการคูณ

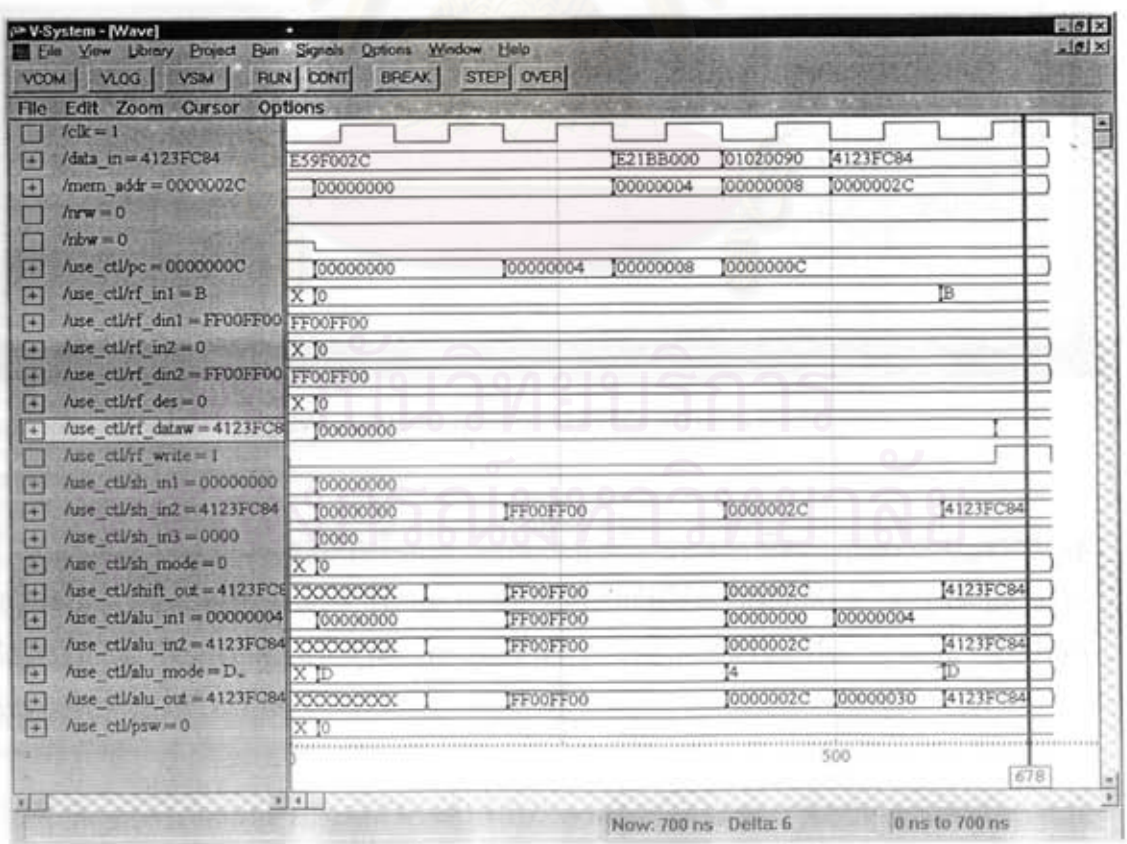


รูปที่6. 3 Waveform ของหน่วยการคูณ

จากรูปที่ 6.3 เป็นการแสดงการคูณค่า FFFF กับ 20000 โดยผลลัพธ์ที่ได้จะเป็นค่า FFFE0000 ดังแสดงในค่าของ mul_out

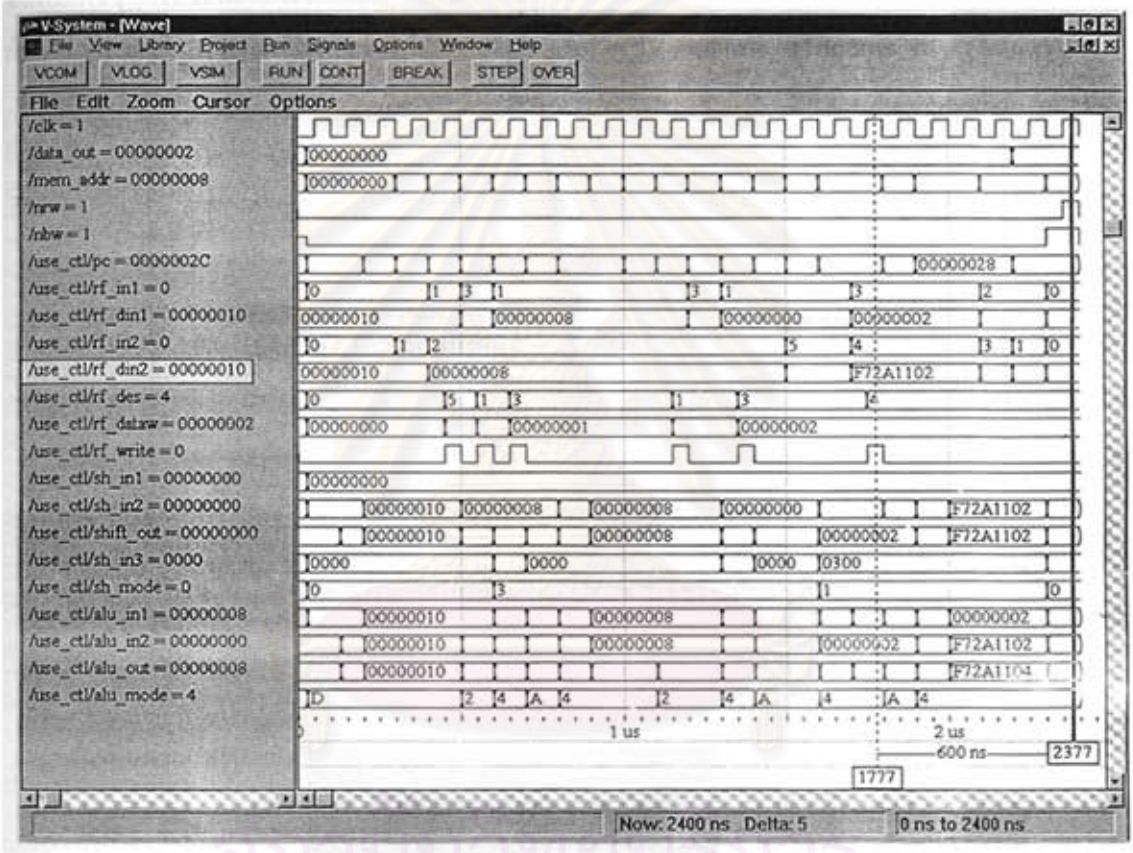
4. การทดสอบคำสั่งอย่างต่อเนื่อง

เพื่อตรวจสอบความถูกต้องของ อาร์ม 7 ที่ได้ออกแบบนั้นมีการทำงานที่ถูกต้องหรือไม่ ดังนี้



รูปที่6. 4 Waveform ของการทำงานคำสั่งโหลด

จากรูปที่ 6.4 เป็นการทวนสอบการทำงานของการทำงานของคำสั่งนำข้อมูลจากหน่วยความจำมาเก็บไว้ในรีจิสเตอร์ตัวที่ 11 ในที่นี้ข้อมูลที่นำเอาเข้ามาคือ 4123FC84 สังเกตที่สัญญาณ data_in ในที่นี้ต้องการนำเข้าไปในรีจิสเตอร์ที่ 11 คือ B สังเกตได้ค่า rf_in1 จะมีค่าเป็น B ตรงตามที่ต้องการและพอร์ทของการเขียนข้อมูลลงสู่รีจิสเตอร์ (rf_dataw) ก็มีค่าเช่นเดียวกับค่าใน data_in คือ 4123FC84 พร้อมทั้งสัญญาณของการเขียนค่าก็ถูกเซตตามสัญญาณของ rf_w โดยการทำงานในส่วนนี้จะมีการเอาค่าลงสู่บัสข้อมูลแล้วผ่านค่าทางหน่วยเลื่อนค่าโดยจะสังเกตพบว่าทำงานได้ถูกต้อง จากค่าที่ออกจากหน่วยเลื่อนค่าคือ shift_out

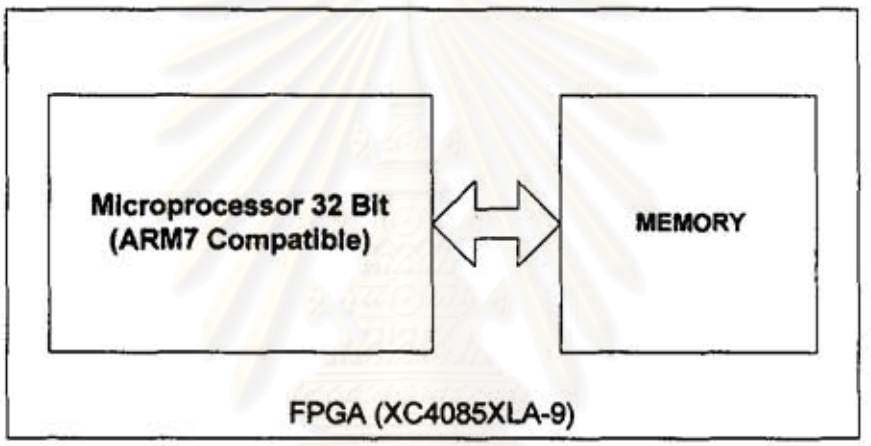


รูปที่6. 5 Waveform ของการทำโปรแกรมการหาร

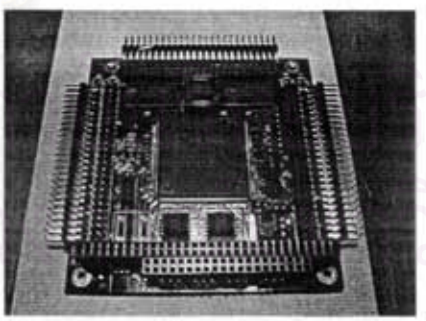
จากรูปที่ 6.5 เป็นผลจากการเขียนโปรแกรมการหาร โดยการลดค่าทีละ 8 $(00000008)_{16}$ จากค่า 16 $(00000010)_{16}$ นั้นหมายถึงการเอา 8 ไปหาร 16 นั้นเองผลลัพธ์ที่ได้จะต้องเป็นค่า 2 โดยจะเขียนค่าลงในรีจิสเตอร์ที่ 4 โดยดูจากค่าใน rf_des ผลลัพธ์ที่ได้สามารถสังเกตได้จากค่าใน rf_dataw ซึ่งเป็นค่า 00000010

ผลจากการจำลองการทำงานด้วยชุดคำสั่งในลักษณะของชุดทดสอบ ผลปรากฏว่าสามารถตรวจสอบข้อผิดพลาดที่เกิดจากคาบเวลาที่ใช้ผิดพลาด และคำสั่งของการ load/store ผิดพลาดจึงได้ทำการแก้ไข

แล้วใช้ชุดทดสอบเดิมตรวจสอบการทำงานอีกครั้งจนกระทั่งไม่พบข้อผิดพลาดใดเกิดขึ้นในการจำลองการทำงาน สิ่งที่พบได้ชัดเจนจากการจำลองการทำงานคือสามารถใช้ค่าหน่วงเวลาของเกตใน Xilinx เอฟพีจีเอที่อยู่ในลักษณะคลังโปรแกรม เพื่อมาเพิ่มการทดสอบค่าคงที่ของเวลาในการจำลองการทำงานให้คล้ายการทำงานบนเอฟพีจีเอมากที่สุด ดังนั้นจึงสามารถเพิ่มความมั่นใจในโมดูลที่ออกแบบได้ว่าเมื่อโปรแกรมลงบนเอฟพีจีเอแล้วจะไม่พบข้อผิดพลาดที่เกิดจากการทำงานของวงจรที่ออกแบบ โดยในการทำชุดต้นแบบโดยใช้เอฟพีจีเอแทนการทำงานของไมโครโพรเซสเซอร์ 32 บิตที่ออกแบบ ทางผู้วิจัยได้เลือกใช้งานเอฟพีจีเอเบอร์ XC4085XLA-9 ซึ่งมีขนาดสามารถรองรับวงจรลอจิกได้ถึง 85,000 เกต โดยไมโครโพรเซสเซอร์ที่ออกแบบมีขนาดประมาณ 31,000 เกต ส่วนเกตที่ว่างได้ออกแบบให้เป็นหน่วยความจำเก็บเพื่อชุดคำสั่งที่ต้องการทดสอบ เพื่อความสะดวกและมีประสิทธิภาพมากกว่าการสร้างบอร์ดติดต่อกับหน่วยความจำภายนอก ซึ่งอาจก่อให้เกิดสัญญาณรบกวนขึ้นในการทดสอบได้ง่าย



รูปที่6.6 การทดสอบไมโครโพรเซสเซอร์ด้วยเอฟพีจีเอ



รูปที่6. 7 บอร์ดต้นแบบด้วย XC4085XLA

จากรูปที่ 6.7 แสดงบอร์ดต้นแบบของไมโครโพรเซสเซอร์ 32 บิตอ้างอิงการทำงานตามอาร์ม 7 ในส่วนของโหมดการทำงานปกติ ซึ่งใช้เอฟพีจีเอเบอร์ XC4085XLA-9 มีขาทั้งหมด 240 ขา และสามารถติดต่อกับคอมพิวเตอร์ส่วนบุคคลได้ทาง ISA Bus โดยเป็นรูปแบบพิเศษบนบอร์ดทดสอบที่ได้สั่งทำ เอฟพีจีเอที่ใช้จะมีการทำงานที่ 3.3 V ไมโครโพรเซสเซอร์ที่ออกแบบ ได้ทดสอบการทำงานใน

ความเร็วที่ 500 KHz และตรวจจับการเปลี่ยนแปลงค่าของบัส ในแต่ละคำสั่งด้วย Logic Analyzer การทำงานของบอร์ดเอฟพีจีเอเริ่มจากการโหลดลอจิกที่ได้จากการออกแบบลงในเอฟพีจีเอ เพื่อให้ทำงานแทนไมโครโพรเซสเซอร์ และส่วนหนึ่งของเนื้อที่บนเอฟพีจีเอ จะเป็นส่วนของหน่วยความจำที่เก็บคำสั่งที่จะใช้ในการทดสอบ ในที่นี้ได้จากชุดทดสอบชุดเดิมที่ได้ทดสอบด้วยการจำลองการทำงาน จากนั้นไมโครโพรเซสเซอร์จะเฟสคำสั่งที่เก็บในส่วนของหน่วยความจำเข้าประมวลผลทั้งหมด 600 ชุดทดสอบ ซึ่งเป็นชุดเดียวกับที่ใช้จำลองการทำงาน โดยสามารถตรวจการทำงานได้โดยการใช้ Logic Analyzer ตรวจจับสัญญาณที่มีการเปลี่ยนแปลง พบว่าปัญหาหรือข้อผิดพลาดที่เกิดขึ้นสามารถสรุปได้ดังตารางที่ 6.10

ปัญหาที่พบ	จำนวนครั้งที่ผิดพลาด
การทำงานของ FPGA	12
หน่วยความจำ	9
การทำงานของไมโครโพรเซสเซอร์ผิดพลาด	3

ตารางที่6. 10 ปัญหาที่พบในการทดสอบบอร์ด

จากการใช้ชุดทดสอบเพื่อทำการบนบอร์ดทดสอบเอฟพีจีเอได้ผลการทำงานดังตารางที่ 6.10 ซึ่งแสดงถึงความผิดพลาดส่วนมากที่เกิดจากสิ่งแวดล้อมซึ่งไม่ใช้สาเหตุจากวงจรที่ออกแบบ พบข้อผิดพลาดของไมโครโพรเซสเซอร์ 3 ครั้งอันเกิดจากการทำงานผิดพลาดของลอจิกส่วนที่เป็นการเชื่อมต่อกับหน่วยความจำที่สร้างขึ้นพิเศษ เพื่อใช้ในการติดต่อกับ SRAM ภายนอกโดยไม่เกี่ยวกับวงจรภายในที่เป็นการทำงานของไมโครโพรเซสเซอร์อาร์ม 7 ส่วนข้อผิดพลาดที่เกิดจากการทำงานของเอฟพีจีเอพบว่าเกิดจากค่าแรงดันของสัญญาณไม่เพียงพอทำให้ไม่สามารถตรวจจับสัญญาณที่เกิดขึ้น นอกจากนี้เกิดจากการผิดพลาดของการโปรแกรมไมโครโพรเซสเซอร์ที่ทำการออกแบบลงสู่เอฟพีจีเอ ซึ่งเป็นผลมาจากสัญญาณการควบคุมขาของเอฟพีจีเอผิดพลาด สุดท้ายข้อผิดพลาดที่เกิดจากหน่วยความจำเป็นข้อผิดพลาดที่เกิดจากบอร์ดเชื่อมต่อมีการออกแบบการอ้างอิงเลขที่อยู่ผิดพลาด รวมทั้งการต่อเชื่อมสายบนบอร์ดผิด

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย