บทที่ 5

การทวนสอบไมโครโปรเซสเซอร์ เออาร์เอ็ม 7

การทวนสอบการทำงานของไมโครโปรเซสเซอร์เป็นงานที่สำคัญอย่างหนึ่ง เพื่อเป็นการตรวจความถูกต้องของการทำงานของระบบที่ออกแบบก่อนที่จะเข้าสู่ขั้นตอนการผลิตเนื่องจากค่าใช้จ่ายในกระบวนการเจือสารสูงมาก ดังนั้นจึงต้องมั่นใจว่าวงจรที่ออกแบบมีความถูกต้องสมบูรณ์มากที่สุด โดยในที่นี้การทวนสอบไมโครโปรเซสเซอร์ใช้วิธีการทวนสอบโดยการจำลองการทำงาน(Simulation-based Verification) การทวนสอบในลักษณะนี้ นักออกแบบจำเป็นต้องใช้การเขียนชุดทดสอบ (Test vectors) เพื่อใช้ในการทวนสอบขึ้นมาเอง งานวิจัยนี้จะมีการเก็บคำสั่งที่ต้องการทดสอบอยู่ในลักษณะของหน่วยความจำ เพื่อให้ไมโครโปรเซสเซอร์สามารถเรียกคำสั่งเข้าไปทำงาน และแสดงผลลัพธ์ที่ได้ โดยการเขียนการทวนสอบในลักษณะของ Test Bench ด้วยภาษา VHDL เข้าไปทำงานในซอฟต์แวร์ที่ใช้ในการจำลองการทำงานระบบทั้งหมดทั้งในระดับของอาร์ทีแอลและการทวนสอบในระดับของเกตกับคุณลักษณะที่กำหนด โดยใช้ค่าคงที่ของเวลาที่ใช้จริงในแต่ละเกตบนเอฟพีจีเอ แต่ทั้งนี้การสร้างคำสั่งด้วยภาษา VHDL ในระดับอาร์ทีแอลได้มีการทดสอบการทำงานในแต่ละชิ้นส่วนย่อยโดยมีรายละเอียดในการทดสอบในส่วน ถัดไป เพื่อให้มีความถูกต้องของแต่ละชิ้นส่วนย่อยก่อนที่จะมีการรวมเป็นระบบแล้วทวนสอบในระดับอาร์ทีแอล หรือเกตต่อไป

5.1 การทดสอบแต่ละชิ้นส่วนในขั้นตอนการออกแบบ

5.1.1 ALU Testing การทดสอบ ตัวคำนวณและตรรกะ กระทำโดยการจำลองการทำงานของตัวคำนวณและตรรกะด้วยชุดทดสอบที่มีการสั่งงานการทำงานทางลอจิกและคณิตศาสตร์ให้กับ ตัวคำนวณและตรรกะด้วยจะกำหนดตัวถูกดำเนินการของในแต่ละคำสั่งไว้ด้วย ดังนั้นจึงสามารถทราบและคำนวณได้ ค่าผลลัพธ์ที่ถูกต้องจะถูกเก็บใส่ใน Test Bench เพื่อเปรียบเทียบผลที่ได้จากการจำลองการทำงานกับค่าผลลัพธ์ที่ถูกต้อง ในการตรวจสอบหน่วยคำนวณและตรรกะจะคิดกรณีที่ตัวบ่งชี้เกิดการเปลี่ยนแปลงด้วย

5.1.2 Shifter Testing เป็นการจำลองการทำงานเช่นเดียวกับการทดสอบ หน่วยคำนวณและตรรกะชุดทดสอบที่ใช้ทดสอบจะเป็นการกำหนดให้ ตัวเลื่อนค่ามีการทำงานตามฟังก์ชันทั้งหมดที่มี โดยมีการสร้างค่า ตัวถูกดำเนินการ เพื่อป้อนให้กับ เลื่อนค่าด้วย ทั้งจะมีตรวจสอบ ตัวบ่งชี้ ที่เกิดจากการ เลื่อนค่า ด้วย

5.1.3 Multiplier Testing เป็นการทดสอบการคูณในที่นี้จะสมมติค่าเพื่อเป็นตัวตั้งและตัวคูณให้กับ ตัวคูณ และตรวจสอบคาบเวลาที่ใช้งาน

5.1.4 Datamem Testing ทดสอบโดยการกำหนดชุดคำสั่งที่มีในหน่วยความจำและให้ Datamem รับค่าข้อมูลเข้าไป ดูค่าที่ได้จากสัญญาณในการจำลองการทำงาน

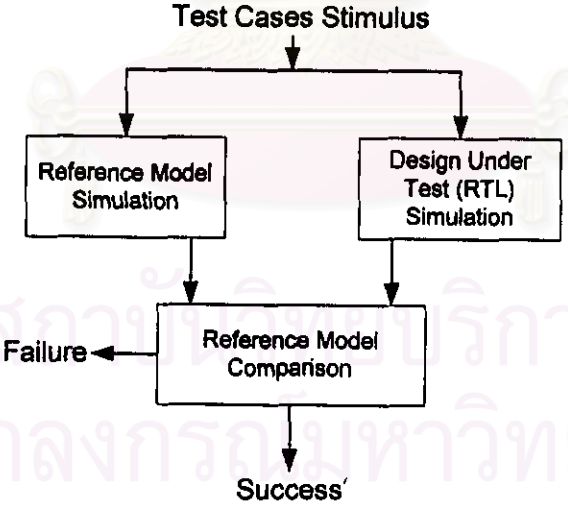
5.1.5 Mar Testing ในส่วนนี้จะมีการทดสอบไปพร้อมๆกับการทำงานทั้งระบบ เนื่องจากจะต้องมีการนำคำสั่งเข้ามาทำงานแล้วให้มีการเปลี่ยนแปลงค่าของ พีซี นอกจากนี้ยังตรวจโดยการให้มีใช้คำสั่งกระโดดข้ามการทำงานเพื่อตรวจสอบค่า พีซี ด้วย รวมถึงคำสั่งที่ใช้ในการติดต่อหน่วยความจำที่ต้องใช้ค่าพีซี

5.1.6 Register File Testing ทำการทดสอบด้วยการจำลองการทำงาน ให้มีการเขียนและอ่านค่าจากรีจิสเตอร์

5.1.7 Control Unit Testing จะเป็นการทดสอบระบบด้วยรวมทั้งหมด ให้สามารถทำงานได้สอดคล้องและมีความต่อเนื่องของการทำงาน โดยการป้อนชุดคำสั่งเพื่อให้ไมโครโปรเซสเซอร์ทำงานทีละหลายๆคำสั่ง และตรวจผลลัพธ์ที่ได้

5.2 การทวนสอบโดยการจำลองการทำงานและสร้างชุดทดสอบ (Simulation-based Verification and Test Cases Generation)

การทวนสอบแบบจำลองการทำงานเป็นการทวนสอบที่สะดวกและนิยมใช้กัน แต่การทวนสอบที่มีประสิทธิภาพจำเป็นต้องสร้างชุดทดสอบที่น้อยแต่สามารถครอบคลุมชุดคำสั่งได้มากที่สุดเพื่อประหยัดเวลาในการจำลองการทำงาน โดยทั่วไปกระบวนการทวนสอบแบบจำลองการทำงาน จะแสดงได้ดังรูปที่ 5.1



รูปที่ 5.1 ขั้นตอนของการทวนสอบด้วยการจำลองการทำงาน

Test cases Stimulus : เป็นชุดทดสอบที่ถูกเลือกเป็นตัวแทนของคำสั่งการทำงานทั้งหมดของไมโครโปรเซสเซอร์ เออาร์เอ็ม 7

Reference Model Simulation : แสดงถึงคุณลักษณะที่ต้องการ (Specification) เพื่อเป็นข้อมูลอ้างอิง ในที่นี้ใช้ Development Debugger ของ เออาร์เอ็ม 7

Design Under Test (RTL) : เป็นส่วนของวงจรที่ออกแบบ(Implementation) ในที่นี้คือไมโครโปรเซสเซอร์ เออาร์เอ็ม 7 อธิบายการทำงานด้วย VHDL

Reference Model Comparison : เป็นส่วนของการเปรียบเทียบผลลัพธ์ที่ได้จากทั้งคุณลักษณะและการวงจรที่ออกแบบ เขียนเป็นชุดทดสอบได้ด้วยภาษา VHDL

5.2.1 การสร้างชุดทดสอบ

การสร้างชุดทดสอบเพื่อการทวนสอบ จะทำการคัดเลือกคำสั่งโดยการวิเคราะห์พฤติกรรมการส่งผ่านข้อมูลระหว่างโมดูลในไมโครโปรเซสเซอร์ เออาร์เอ็ม 7 ในระดับของคำสั่งการทำงาน ซึ่งมีทั้งหมด 2,660 คำสั่งจากคุณลักษณะที่กำหนดไว้ก่อนการออกแบบ โดยแบ่งออกเป็น 2 ชุดทดสอบดังนี้

5.2.1.1 ชุดทดสอบแรก เป็นชุดทดสอบที่จัดแบ่งตามคำสั่งของ เออาร์เอ็ม 7 ทั้งหมด 26 คำสั่งดังตารางที่ 5.1 เนื่องจากคำสั่งของเออาร์เอ็ม สามารถกำหนดให้ไม่มีการคำนวณค่าตัวบ่งชี้ (flag) ดังนั้นชุดทดสอบนี้จึงไม่คำนึงผลของตัวบ่งชี้ที่เกิดขึ้นจากการทำงาน สามารถแบ่ง 26 คำสั่งการทำงานออกได้เป็น 16 คำสั่งใน หน่วยคำนวณและตรรกะ โดยแต่ละคำสั่งจะแบ่งออกได้ 4 กรณีที่แบ่งตามการใช้ตัวถูกดำเนินการดังนี้ มาจากรีจิสเตอร์ทั้งคู่, มาจากรีจิสเตอร์และค่าจำนวนเต็ม, ตัวถูกดำเนินการมาจากรีจิสเตอร์และอีกค่าทำการเลื่อนค่าซึ่งได้ค่ามาจากรีจิสเตอร์, ตัวถูกดำเนินการ มาจากรีจิสเตอร์แต่อีกค่าที่สามารถทำการเลื่อนค่าได้มาจากค่าจำนวนเต็ม ตัวอย่างเช่นการทดสอบการทำงานของคำสั่ง AND จะสามารถใช้ชุดทดสอบ 4 ชุดแทนการทำงานที่เกิดขึ้นทั้งหมดของ AND ได้ดังตารางที่ 5.2 การคำนึงถึงชนิดการใช้ตัวถูกดำเนินการเนื่องจากการเรียกใช้ตัวถูกดำเนินการที่แตกต่างกันจะมีการไหลของข้อมูลต่างกัน ซึ่งสามารถมีได้ถึง 4 แบบ ดังนั้นการเลือกชุดทดสอบที่ให้สามารถผ่านไปตามเส้นทางการไหลของข้อมูลและสามารถครอบคลุมการเรียกใช้สัญญาณควบคุมทั้งหมดอีกด้วย

คำสั่งการกระโดดข้ามการทำงาน (Branch) มีชุดทดสอบทั้งในกรณีที่มีการคืนค่ากลับตำแหน่งเดิมได้ (BL) และไม่สามารถกลับไปตำแหน่งเดิมได้ (B) โดดทั้ง 2 แบบจะมีชุดทดสอบการทำงานของไปป์ไลน์ซึ่งจะเกิดกรณีของการ Stalling ไปป์ไลน์ และค่าของ PC ให้มีการทำงานที่ถูกต้อง ดังนั้นเมื่อใช้ชุดทดสอบการทำงานที่มีคำสั่ง B จะทำให้สามารถครอบคลุมการทดสอบทั้งในส่วนของการอ้างอิงหน่วยความจำ, การทำงานไปป์ไลน์ และการทำงานของชุดคำสั่งทั้งหมด

ชุดทดสอบมีคำสั่งของการทำงานในแบบที่เป็นการคูณตามปกติ (MUL) และการคูณที่สามารถมีการคำนวณการบวกได้ (MLA) คำสั่งของการย้ายข้อมูลระหว่างหน่วยความกับรีจิสเตอร์มีชุดทดสอบที่คำนึงข้อมูลที่มีการแลกเปลี่ยน รวมทั้งค่า PC ที่เปลี่ยนไป คำสั่งการกระโดดข้ามการทำงาน การคูณและการย้ายข้อมูลระหว่างหน่วยความจำและรีจิสเตอร์มีชุดทดสอบในการอ้างอิงหน่วยความจำทั้ง 3 แบบ ณ Indirect, Index และ Base Index ดังตารางที่ 5.3

5.2.1.2 ชุดทดสอบที่สอง เป็นชุดทดสอบที่คำนึงถึงการเปลี่ยนแปลงค่าตัวบ่งชี้ (N : Negative, Z : Zero, C : Carry, V : Over flow) ในการทำงานของหน่วยคำนวณและตรรกะ ดังนั้นชุดทดสอบจะคำนึงถึงกรณีที่สามารถทำให้ค่าตัวบ่งชี้เกิดการเปลี่ยนแปลงดังนี้ ค่า N จะมีการเปลี่ยนแปลงเมื่อบิตสุดท้ายของผลลัพธ์

ตารางที่ 5.1 ชุดคำสั่งทดสอบ 26 ชุด

Instruction	Source 1	Source 2	Shift	Execute	Write-back
B	-	-	-	-	#Jump>Pc
BL	-	-	-	-	#Jump>Pc
AND	W	W	OP2	SHIFT,AND	W
EOR	W	W	OP2	SHIFT,XOR	W
SUB	W	W	OP2	SHIFT,SUB	W
RSB	W	W	OP2	SHIFT,RSB	W
ADD	W	W	OP2	SHIFT,ADD	W
ADC	W	W	OP2	SHIFT,ADC	W
SBC	W	W	OP2	SHIFT,SBC	W
RSC	W	W	OP2	SHIFT,RSC	W
TST	-	-	OP2	SHIFT,TST	-
TEQ	-	-	OP2	SHIFT,TEQ	-
CMP	-	-	OP2	SHIFT,CMP	-
CMN	-	-	OP2	SHIFT,CMN	-
ORR	W	W	OP2	SHIFT,ORR	W
MOV	-	W	OP2	SHIFT,MOV	W
BIC	W	W	OP2	SHIFT,OP2 AND NOT OP1	W
MVN	W	W	OP2	SHIFT,NOT OP2	W
MUL	W	W	-	MUL	W
MLA	W	W	-	MUL,ADD	W
LDR	W,B	W,B	OP2	SHIFT,LOAD	W,B
STR	W,B	W,B	OP2	SHIFT,STORE	W,B
LDM	W,B	W,B	OP2	Multiple Load	W,B
STM	W,B	W,B	OP2	Multiple Store	W,B
SWP	W,B	W,B	OP2	LOAD,STORE	W,B

เป็น 1, Z จะเปลี่ยนแปลงเป็น 1 เมื่อผลลัพธ์ที่ได้มีค่าเป็นศูนย์ทั้ง 32 บิต, C จะเปลี่ยนแปลงเมื่อมีการบวก ลบหรือเลื่อนค่าแล้วผลลัพธ์ที่ได้เกินขนาด 32 บิต, V จะเปลี่ยนแปลงเป็น 1 เมื่อมีการบวก ลบหรือการเปรียบเทียบแบบคิดเครื่องหมายแล้วผลลัพธ์เกินจาก 31 บิต

ตารางที่ 5. 2 แสดงชุดคำสั่งที่ใช้ในการทดสอบส่วน Data processing(ต่อ)

Instruction Set	Instruction	Feature	Description
Data Processing without shift operator	AND	1. register and register 2. register and immediate value	Op1/reg AND op2/reg Op1/reg AND op2/immediate
	EOR	1. register and register 2. register and immediate value	Op1/reg EOR op2/reg Op1/reg EOR op2/immediate value
	SUB	1. register and register 2. register and immediate value 3. op1 little than op2	Op1/reg - op2/reg Op1/reg - op2/immediate Op1 - op2 but op1 little than op2
	RSB	1. register and register 2. register and immediate value 3. op1 larger than op2	Op2/reg – op1/reg Op2/reg – op1/immediate Op2 – op1 but op1 larger than op2
	ADD	register and register register and immediate value overflow add	Op1/reg + op2/reg Op1/reg + op2/reg Op1 + op2 but overflow
	ADC	register and register register and immediate value carry = 1	Op1/reg + op2/reg Op1/reg + op2/immediate value Op1 + op2 + carry
data Processing With shift operand	SBC	register and register add carry register and immediate value add carry op1 lower than op2	Op1/reg – op2/reg + C Op1/reg – op2/immediate + C Op1 – op2 + C but op1 lower than op2
	RSC	register and register register and immediate value op2 lower than op1	Op2/reg – op1/reg + Op2/reg – op1/immediate + C Op2 – op1 + C but op2 lower than op1
	TST	1. register 2. immediate value	Set condition code on op1 AND op2

ตารางที่ 5. 3 แสดงชุดคำสั่งที่ใช้ในการทดสอบส่วน Data processing(ต่อ)

Instruction Set	Instruction	Feature	Description
Data Processing	TEQ	1. register 2. immediate value	Set condition code op1 EOR op2
	CMP	1. register 2. immediate value	Set condition code op1 – op2
	CMN	1. register 2. immediate value	Set condition code op1 + op2
	ORR	1. register and register 2. register and immediate value	Op1/reg OR op2/reg Op1/reg OR op2/immediate
	MOV	1. register to register 2. immediate to register	Destination = op2/reg Destination = op2/immediate
	BIC	1. register and register 2. register and immediate	Op1/reg AND (not op2/reg) Op1/reg AND (not op2/immediate)
	MVN	1. register 2. immediate value	Not op2/reg Not op2/immediate
Branch	B		Jump no back track
	BL		Jump with return
Multiply	MUL	1. register and register 2. overflow	Multiply
	MLA	1. register and register 2. overflow	Multiply and accumulator R1= R2*R3+R4
Single Data Transfer	LDR	1. indirect addressing mode 2. index addressing mode 3. base index addressing mode 4. byte and word	Load Single Data from Memory to Register

ตารางที่ 5. 4 แสดงชุดคำสั่งที่ใช้ในการทดสอบส่วน Data processing(ต่อ)

Instruction Set	Instruction	Feature	Description
Single Data Transfer	STR	1. indirect addressing mode 2. index addressing mode 4. base index addressing mode 3. byte and word	Store Single Data from Register to Memory
Block Data Transfer	LDM	1. pre-increment load 2. post-increment load 3. post-decrement load 4. pre-decrement load	Load multiple Data from Memory to Register
	STM	1. pre-increment store 2. post-increment store 3. post-decrement store 4. pre-decrement store	Store multiple Data from Register to Memory
Data Swap	SWP	Byte and word	Swap data between Register & Memory

Access All 16 Register , R0-R13, R14(Stack), R15(PC)

ลักษณะของการป้อนคำสั่งเมื่อจำลองการทำงานจะเป็นแบบต่อเนื่องกันไปทั้งหมด ซึ่งมีการผสม โดยไม่ได้เรียงลำดับ หรือจัดหมวดหมู่ของแต่ละกลุ่มคำสั่ง ดังนั้นการทำงานจะเหมือนจริงและผสมกันไประหว่างแต่ละกลุ่มคำสั่งเพื่อตรวจสอบความต่อเนื่องในการทำงานของไมโครโพรเซสเซอร์ด้วย ข้อมูลที่ทำการสังเกตและตรวจสอบอย่างละเอียดคือค่าของข้อมูล ค่าพีซีที่เก็บตำแหน่งเลขที่อยู่ ค่าของสัญญาณ รวมถึงคาบของเวลาในการใช้งาน ทั้งนี้การทำงานของไมโครโพรเซสเซอร์อาร์มเป็นแบบมีไปป์ไลน์ ดังนั้นการป้อนชุดคำสั่งอย่างต่อเนื่องจะทำให้สามารถตรวจสอบการทำงานของไปป์ไลน์ได้อีกทางหนึ่ง แต่ไม่สามารถยืนยันได้ว่ามีความถูกต้องในทุกกรณีของการทำงาน เพราะไม่ได้ทำการทดสอบลำดับทุกกรณีที่สามารถเป็นไปได้