รหัสนักศึกษา ๖๖๐๑๐๔๕๖ รหัสนักศึกษา ๖๖๐๑๐๖๖๖

การทดลองที่ A

การทำงานของแคชชนิด Direct Mapped

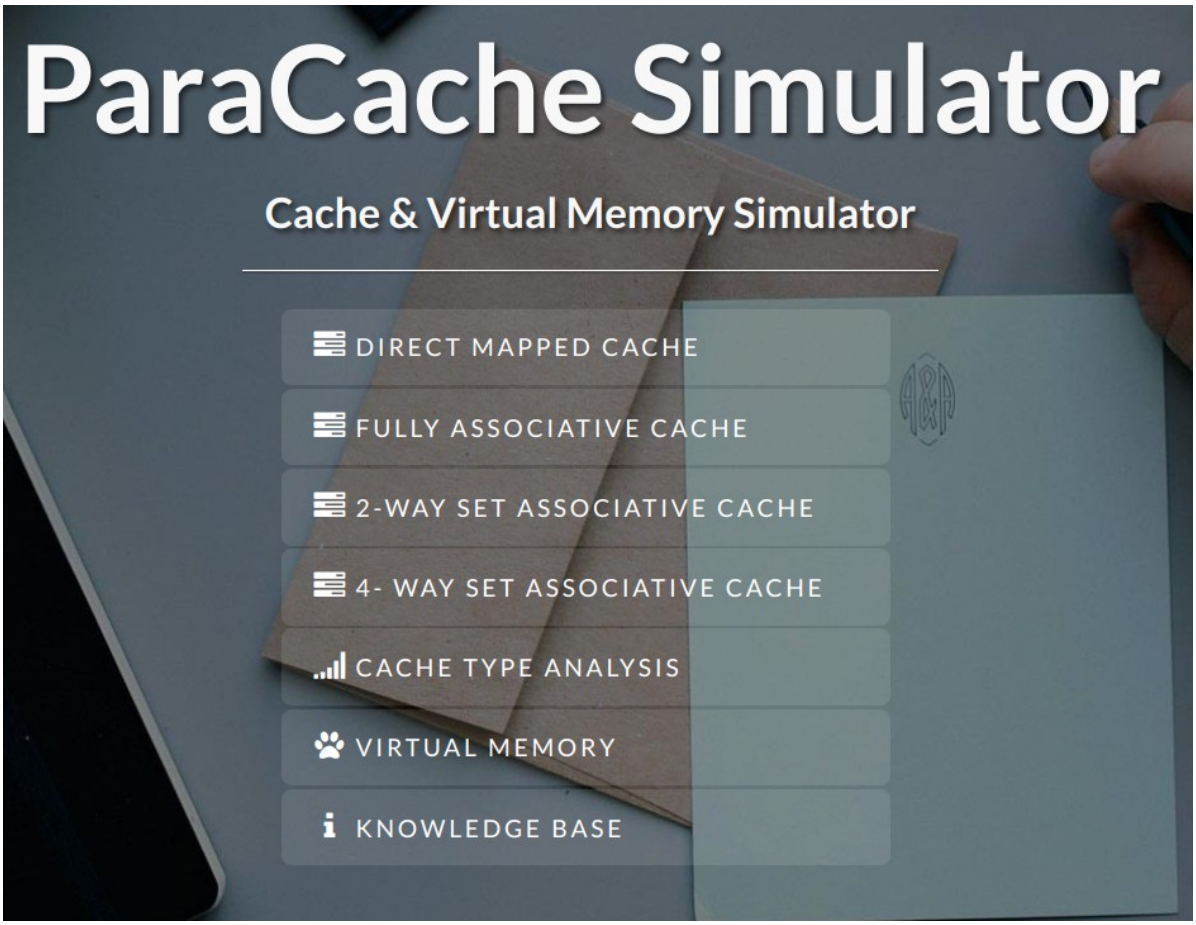
วิชา Computer Organization and Assembly Language

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใช้เว็บเบราส์เซอร์เปิดใช้งานซิมูเลเตอร์ ชื่อ Para Cache

https://www3.ntu.edu.sg/home/smitha/ParaCache/Paracache/start.html



เอกสารอธิบาย

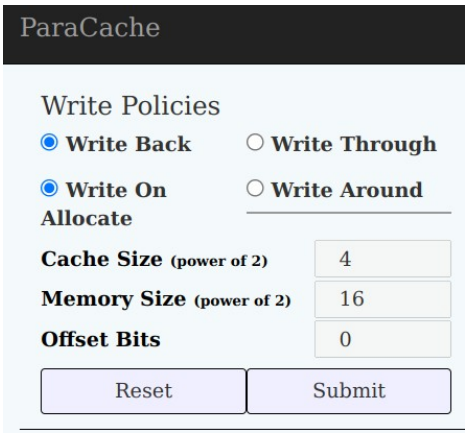
https://www3.ntu.edu.sg/home/smitha/ParaCache/Paracache/kb.pdf

ทำการทดลอง ตามขั้นตอนต่อไปนี้

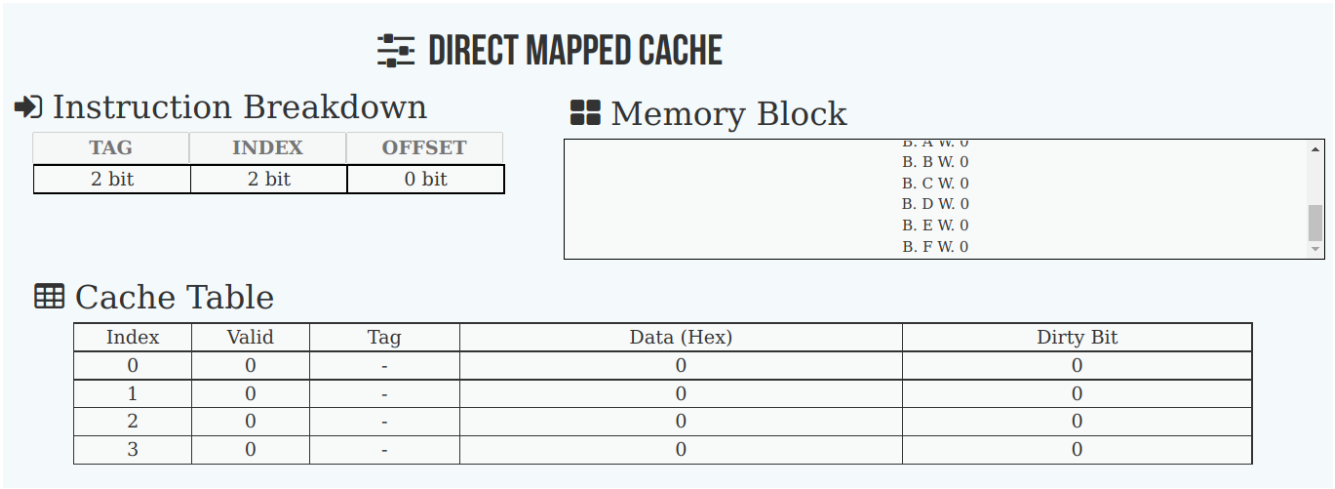
รหัสนักศึกษา 6501043 รหัสนักศึกษา 6501063

1. การทดลอง Direct Mapped Cache

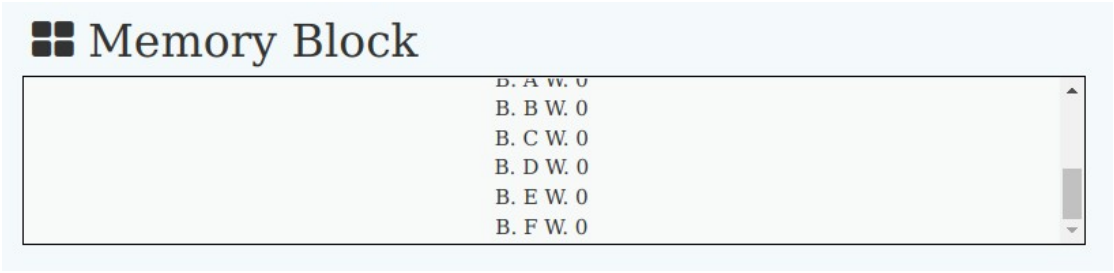
กดเมนู เลือก Direct Mapped Cache ตั้งขนาดและ Write Policy ของแคช ดังรูป



2. กด Submit แล้วสังเกตรายละเอียดของแคชที่อยู่ด้านขวา



เลื่อนหน้าต่างลงไปด้านล่างสุดของ Memory Block โปรดสังเกตหมายเลขบล็อก (B.) มีค่าเท่ากับ 0 ถึง F และหมายเลขเวิร์ด (W.) เท่ากับ 0 เสมอ



เพราะเหตุใด

เลขฐาน offset bit ตัวไม่ได้ 0 ซึ่งมีค่าออกมาไม่ค่า 0 ตลอดเลย B ขึ้นอยู่กับ memory size

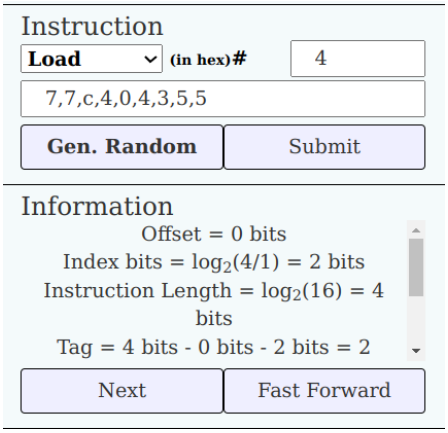
W. ขึ้นอยู่กับ offset bit

รหัสนักศึกษา 6501048 รหัสนักศึกษา 6501063

3. การทดลองคำสั่ง Load Instruction ที่หมายเลขแอดเดรสที่ต้องการ หรือ ให้โปรแกรมสุ่มหมายเลขแอดเดรสให้

กรอก 4 ลงในหมายเลขฐานสิบหกที่มีอยู่ในกล่องข้อความด้านขวา

กรอกหมายเลข 7, 7, c, 4, 0, 4, 3, 5, 5 ในกล่องข้อความดังรูป



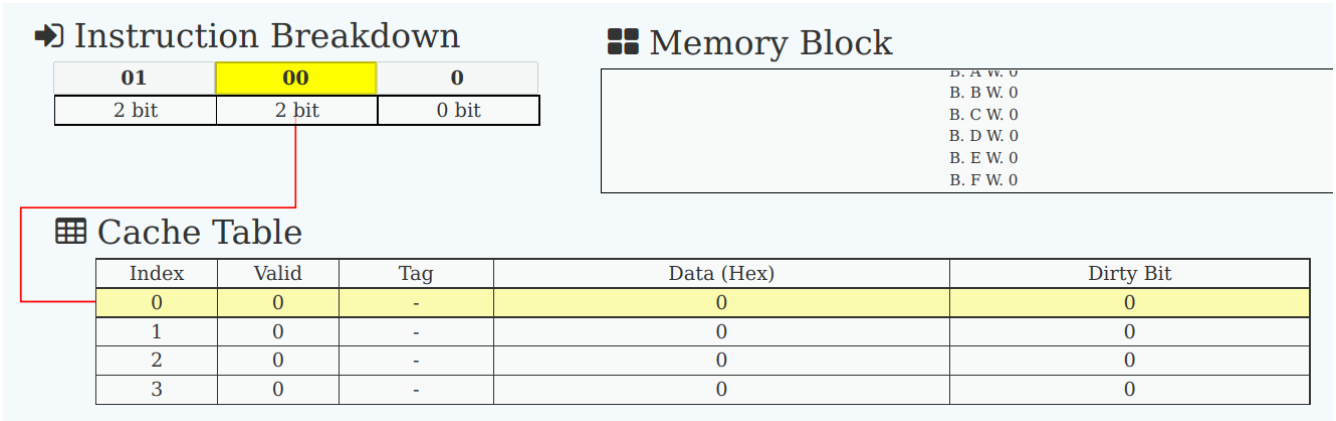
อธิบาย information ในรูปว่า Tag, Index และ Offset สัมพันธ์กับ Cache Size และ Memory Size ที่กรอก

Cache size จะมีผลต่อ Index memory size มีผลต่อ Instruction length ซึ่งเอา 2 ตัวนี้ ไปคำนวน tag

4. กดปุ่ม Submit หมายเลข 4 ที่กรอก โปรดสังเกตและอ่านกล่องข้อความที่เป็นสีชมพู อธิบายตามความเข้าใจ

4 จะถูกเปลี่ยนเป็น binary จาก Hex แล้วแบ่งให้ tag, index & offset

5. กดปุ่ม Next และสังเกตกล่องข้อความที่เปลี่ยนเป็นสีเหลืองว่าเกี่ยวข้องกับหมายเลขที่ Submit ไปก่อนหน้านี้อย่างไร

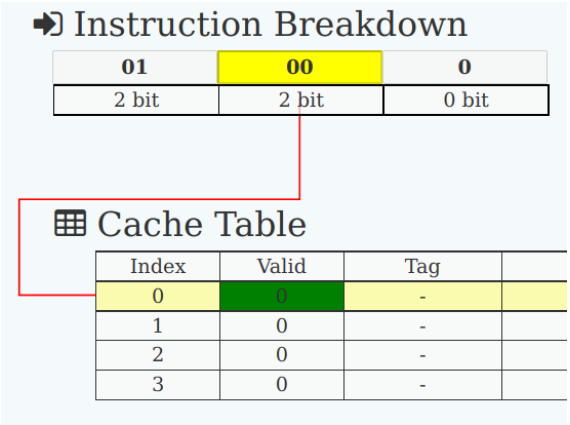


อธิบายความสัมพันธ์ระหว่าง Instruction Breakdown 01 00 และหมายเลข 4

0100 คือ 4 ในเลขฐาน 2 และแบ่งออกให้ tag, index

รหัสนักศึกษา 6301048 รหัสนักศึกษา 6301611

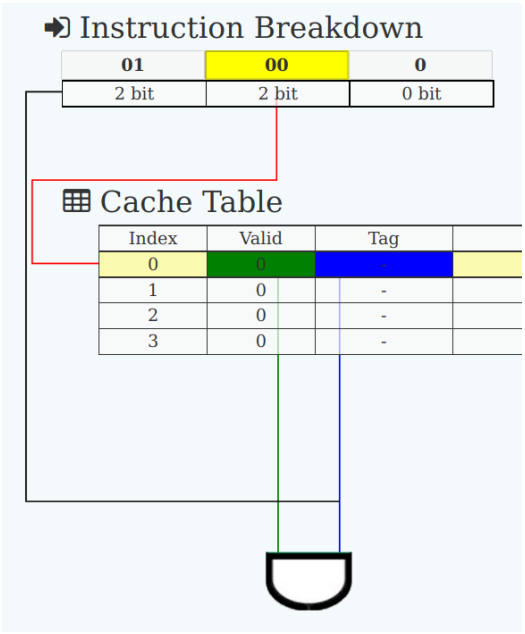
6. กดปุ่ม Next และสังเกตกล่องข้อความที่เปลี่ยนเป็นสีเขียว



อธิบายรูปนี้ และบิต Valid จึงเป็น 0

ไม่มีค่า อะไรถูก เก็บไว้ใน cache

7. กดปุ่ม Next และสังเกตกล่องข้อความที่เปลี่ยนเป็นน้ำเงิน และ AND เกตว่าทำกระบวนการอะไรกัน

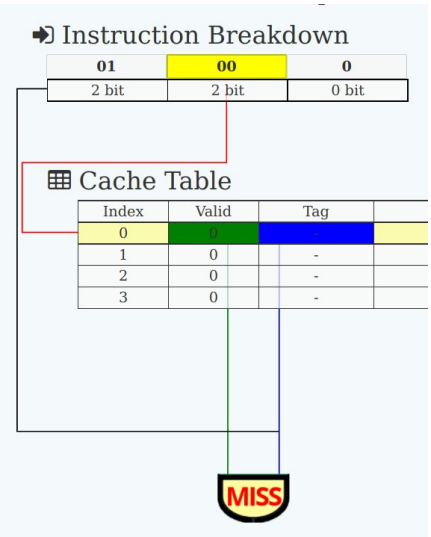


อธิบายว่า Tag จึงมีสัญลักษณ์ '-'

ยังไม่มีค่า อะไรอยู่ข้างใน cache

รหัสนักศึกษา 6401041 รหัสนักศึกษา 6401063

8. กดปุ่ม Next ต่อเพื่อดำเนินการต่อ โปรดสังเกตข้อความบน AND เกต



กดปุ่ม Next เพื่อดำเนินการต่อ โปรดสังเกต Cache Table ว่ามีการเปลี่ยนแปลงอย่างไร

Cache Table

Index	Valid	Tag	Data (Hex)	Dirty Bit
0	1	01	BLOCK 4 WORD 0 - 0	0
1	0	-	0	0
2	0	-	0	0
3	0	-	0	0

อธิบายบิต Valid Tag และ Data (Hex) จึงเปลี่ยนเป็นรูปนี้

Valid ของช่อง index ที่ 0 มีค่าเป็น 1 , data คือ ของมัน เก็บต่างจาก block ที่ 4 ใน main memory

9. กดปุ่ม Next เพื่อดำเนินการต่อ โปรดสังเกตข้อมูลสถิติสีเหลืองด้านล่าง

Statistics

Hit Rate : 0%

Miss Rate : 100%

List of Previous Instructions :

- **Load 4 [Miss]**

อธิบายข้อมูลที่ได้

ค่าใน cache ไม่มี อันไหนตรงกับที่ request เลย หรือ miss หมดเลย จากนั้นเก็บค่าใส่ใน cache

10. โปรดสังเกตหมายเลขแอดเดรสถัดไปจะย้ายมาในกล่องข้อความด้านขวาบนของรูปนี้ กดปุ่ม Submit

11. กดปุ่ม Fast Forward เพื่อเร่งการทำงานของคำสั่งให้รวดเร็วขึ้น โปรดสังเกตการเปลี่ยนแปลงใน Cache Table และ Statistics หลังจากนั้น

01	11	0
2 bit	2 bit	0 bit

Memory Block

Cache Table

Index	Valid	Tag	Data (Hex)
0	1	01	BLOCK 4 WORD 0 - 0
1	0	-	0
2	0	-	0
3	1	01	BLOCK 7 WORD 0 - 0

Statistics

Hit Rate : 0%

Miss Rate : 100%

List of Previous Instructions :

- **Load 4 [Miss]**
- **Load 7 [Miss]**

12. กด Submit และ Fast Forward เรื่อยๆ จนไม่เหลือหมายเลขแอดเดรส โปรดสังเกตการเปลี่ยนแปลงใน Statistics หลังจากนั้น

Instruction

Load (in hex)#

List of next 10 Instructions

Gen. Random | Submit

Information

The cycle has been completed.
Please submit another instructions

Next | Fast Forward

รหัสนักศึกษา 6301041 รหัสนักศึกษา 6301041

Statistics

Hit Rate : 20%

Miss Rate : 80%

List of Previous Instructions :

- **Load 4 [Miss]**
- **Load 7 [Miss]**
- **Load 7 [Hit]**
- **Load C [Miss]**
- **Load 4 [Miss]**
- **Load 0 [Miss]**
- **Load 4 [Miss]**
- **Load 3 [Miss]**
- **Load 5 [Miss]**
- **Load 5 [Hit]**

อธิบายข้อมูลที่ได้ว่า Hit Rate และ Miss Rate คำนวณอย่างไร

Hit rate คือ จน.ที่ Hit / จนวน request ทั้งหมด , miss rate คือ 100 - Hitrate หรือ จำนวน miss / จำนวน ที่ request มาทั้งหมด

นักศึกษาควรจะได้ผลการทดลองใน Cache Table ตรงกับรูปนี้

Cache Table

Index	Valid	Tag	Data (Hex)
0	1	01	BLOCK 4 WORD 0 - 0
1	1	01	BLOCK 5 WORD 0 - 0
2	0	-	0
3	1	00	BLOCK 3 WORD 0 - 0

13. กรอกหมายเลขบล็อกที่แคชมีอยู่ เพื่อจงใจให้เกิด แคชฮิต ดังรูป

Instruction

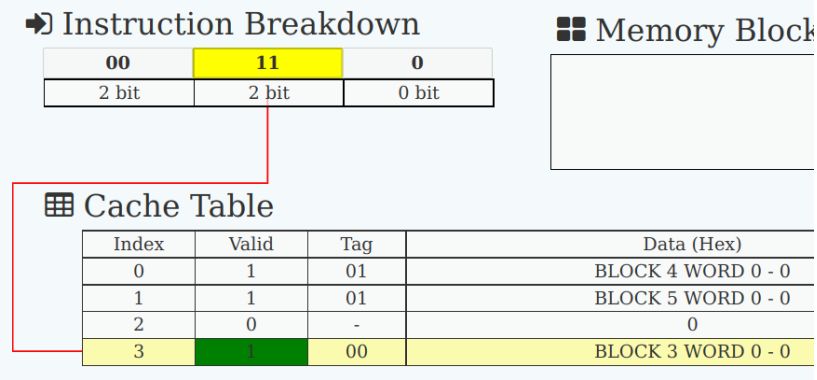
Load **(in hex)#**

3, 4, 5

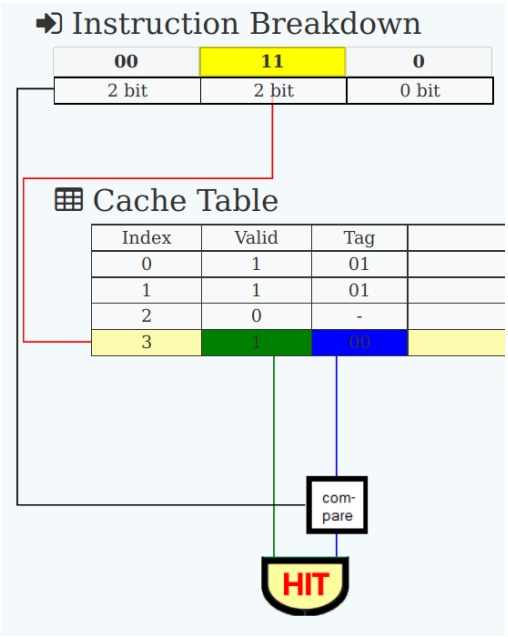
Gen. Random | Submit

กด Submit และ Next จนได้เหตุการณ์นี้

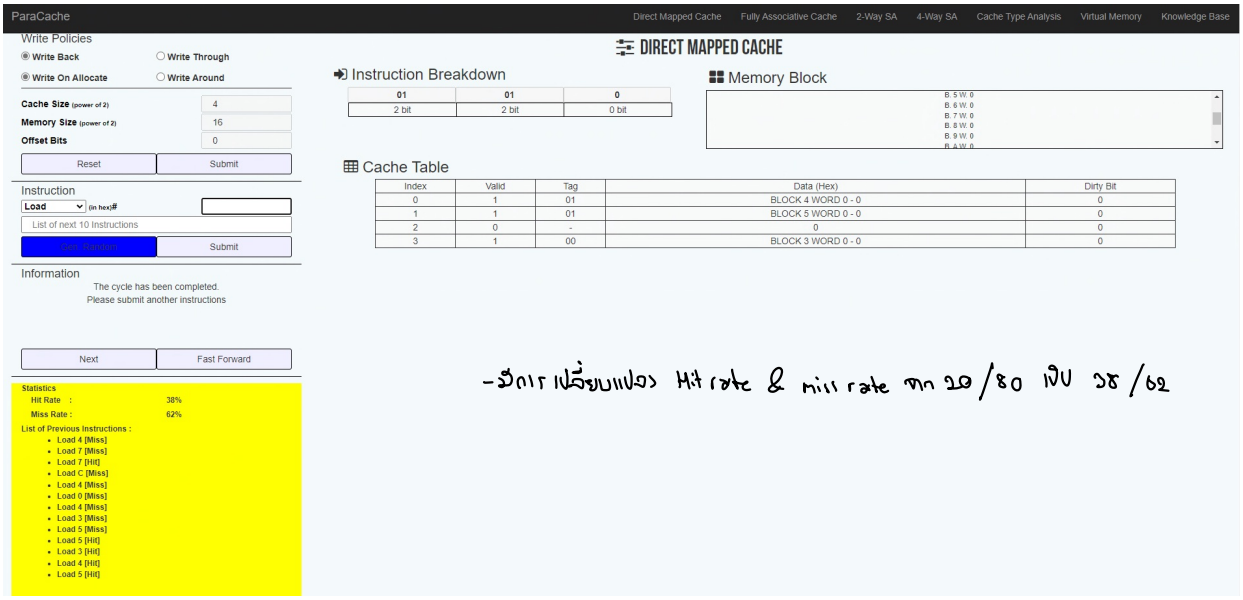
รหัสนักศึกษา 6201048 รหัสนักศึกษา 6201077



โปรดสังเกตคอลัมน์ Valid และ Tag ว่าตรงกันหรือไม่



14. กดปุ่ม Submit หมายเลขถัดไปจนหมด และแนบรูปตาราง Statistics ว่ามีการเปลี่ยนแปลงหรือไม่ อย่างไร

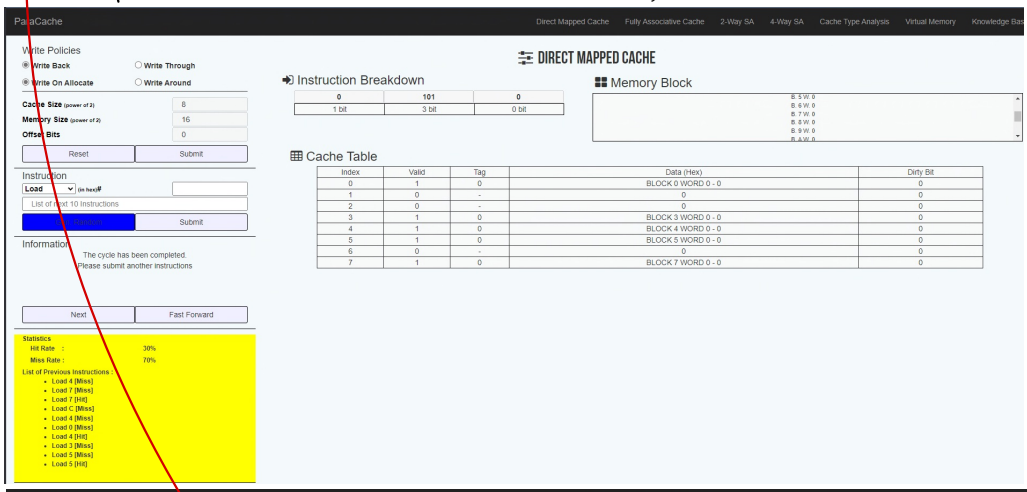


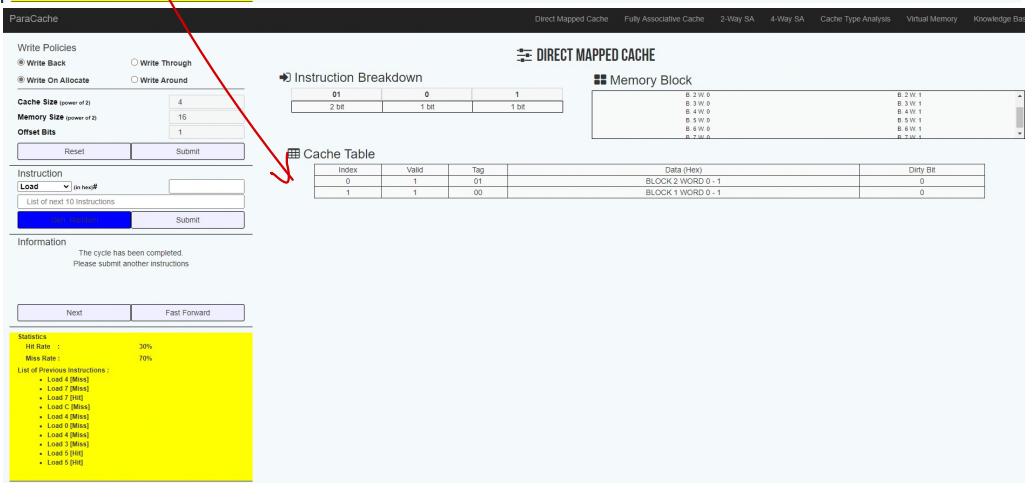
-มีการเปลี่ยนแปลง Hit rate & miss rate จาก 20/80 เป็น 38/62

รหัสนักศึกษา 6701047 รหัสนักศึกษา 6701051

กิจกรรมท้ายการทดลอง

1. ศึกษาการทำงานของ Load Instruction เช่นเดิม
   - ตั้งขนาดของแคชเท่ากับ 8 และ Memory Size เท่ากับ 16 OFFSET = 0 บิต แล้วเปรียบเทียบ
   - ตั้งขนาดของแคชเท่ากับ 4 และ Memory Size เท่ากับ 16 OFFSET = 1 บิต แล้วเปรียบเทียบ
2. ศึกษาการทำงานของ Load Instruction เช่นเดิม แต่ตั้ง Write Policy เป็น Write Through และ Write Around
   - ตั้งขนาดของแคชเท่ากับ 4 และ Memory Size เท่ากับ 16 OFFSET = 0 บิต
   - ตั้งขนาดของแคชเท่ากับ 8 และ Memory Size เท่ากับ 16 OFFSET = 0 บิต
   - ตั้งขนาดของแคชเท่ากับ 4 และ Memory Size เท่ากับ 16 OFFSET = 1 บิต
3. เปลี่ยน Instruction เป็น Store เพื่อศึกษาการทำงานของ Dirty Bit โดยตั้ง Write Policy เป็น Write Back และ Write on Allocate
   - ตั้งขนาดของแคชเท่ากับ 4 และ Memory Size เท่ากับ 16 OFFSET = 0 บิต
   - ตั้งขนาดของแคชเท่ากับ 8 และ Memory Size เท่ากับ 16 OFFSET = 0 บิต
   - ตั้งขนาดของแคชเท่ากับ 4 และ Memory Size เท่ากับ 16 OFFSET = 1 บิต
4. เปลี่ยน Instruction เป็น Store เพื่อศึกษาการทำงานของ Dirty Bit โดยตั้ง Write Policy เป็น Write Through และ Write Around
   - ตั้งขนาดของแคชเท่ากับ 4 และ Memory Size เท่ากับ 16 OFFSET = 0 บิต
   - ตั้งขนาดของแคชเท่ากับ 8 และ Memory Size เท่ากับ 16 OFFSET = 0 บิต
   - ตั้งขนาดของแคชเท่ากับ 4 และ Memory Size เท่ากับ 16 OFFSET = 1 บิต





ต่างกันที่ การตั้งค่า cache, memory size มีผลต่อ index, instruction length, tag ที่ใน cache table แสดงผลลัพธ์ & เก็บค่า cache ได้ต่างกัน