

## CEDT Final Project 2568 Digital Logic 2110252: Mini CPU

ปล่อย Testcase 20 เรียบร้อยครับ

Template สามารถเข้าถึงได้จาก [TEMPLATE Project 2568.dia](#)

Template ทุก Testcase

[https://drive.google.com/file/d/1-lbcwmmF0I7HENbjkT9V\\_YOa5zmdLla8/view?usp=sharing](https://drive.google.com/file/d/1-lbcwmmF0I7HENbjkT9V_YOa5zmdLla8/view?usp=sharing)

Testcase full (Excel)

[https://docs.google.com/spreadsheets/d/1MF2TL-rYU\\_UMsCkPACtszQV-eJuxBxd-/edit?usp=sharing&ouid=115188683733264074142&rtpof=true&sd=true](https://docs.google.com/spreadsheets/d/1MF2TL-rYU_UMsCkPACtszQV-eJuxBxd-/edit?usp=sharing&ouid=115188683733264074142&rtpof=true&sd=true)

คะแนนการรันจะนับถึงเที่ยงคืนวันนี้นะครับ

ตอนนี้ทุก test case มีการทำผ่านครบทั้ง 29 cases และวันนี้ครับ แต่ยังไม่มีกลุ่มไหนทำถูกต้องทั้งหมด  
แจ้งตอน 21:00

มีทำได้เต็ม 3 กลุ่มแล้วนะครับ ข้อมูลตอน 07:00

เปิด grader ด้วยเดิมแล้วนะครับ

มี Test case ของ 2 คำสั่งพิเศษนี้ ให้ใน template เดิมด้วยนะครับ

<https://drive.google.com/file/d/1phE6zkWXZH67jJy4E8BivXvaieF7YzfN/view?usp=sharing>

กลุ่มไหนส่งแล้วผ่านทั้งหมดเรียบร้อย ให้รายงานในหน้าสุดท้ายของไฟล์นี้ด้วยนะครับ

ให้จองลำดับการนำเสนอ (เข้าด้วยอีเมลนิสิต)

[https://docs.google.com/spreadsheets/d/1VMYgTfqbvO5k-T\\_jqUrLrMDFM8EkkyuT6hb0qVfP3TI/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1VMYgTfqbvO5k-T_jqUrLrMDFM8EkkyuT6hb0qVfP3TI/edit?usp=sharing)

- นิสิตต้องส่งรายชื่อสมาชิกผ่าน MCV และถึงจะขึ้นชื่อกลุ่มให้เลือก (อันนี้ไม่ได้อัตโนมัตินะครับ ต้องดึงรายชื่อออกรายชื่อที่ ดังนั้นให้ใส่ข้อมูลให้ครบถ้วนภายใน 09:15 เดียวผมจะดึงข้อมูลให้ช่วงเวลา

นั้น) (<https://www.courseville.com/?q=courseville/worksheet/74041/1793002>)

ให้นิสิตสร้าง CPU โดยใช้โปรแกรม Digital โดยช่วงแรกอรับคำสั่งเพื่อโหลดโปรแกรมเข้ามาเก็บใน RAM memory ขนาด 256 x 14 bits (pRAM - Program RAM) และทำการคำสั่งโดยคำสั่งแรกจะอยู่ที่ address 0x00 ทำไปจนเจอคำสั่งให้หยุด จากนั้นรอจนกว่าจะได้รับสัญญาณให้ส่งคำตอบ จึงแสดงผลลัพธ์ที่เก็บไว้ใน RAM memory ขนาด 256 x 8 bits (rRAM - Result RAM) ในตัวແண່ງ address 0x00 - 0x0F ออกทาง output และ 7-segment จำนวน 2 ตัว

### วงจรมี input คือ

- M ขนาด 8 bits พารามิเตอร์ที่ 1
  - N ขนาด 8 bits พารามิเตอร์ที่ 2
  - progIn ขนาด 14 bits เป็น data input ที่จะส่งโปรแกรมเข้ามาทีละคำสั่งจนครบ ก่อนจะให้รอ สัญญาณเพื่อเริ่มทำงาน
  - reset ขนาด 1 bit เมื่อได้ 1 ให้ reset การทำงานของวงจร และ เคลี้ยรค่าของ rRAM ให้เป็น 0 ทั้งหมด โดยจะมี clock ให้ทำงานส่วนนี้อย่างน้อย 20 clocks
  - progLoad ขนาด 1 bit เมื่อได้ 1 ให้เริ่มอ่านค่าจาก progIn ทีละคำสั่ง ไปเก็บใน pRAM เริ่มตั้งแต่ address 0x00 ไปจนกระทั่ง สัญญาณ progLoad เป็น 0 ซึ่งคำสั่งที่มากที่สุดจะเป็น 256 คำสั่ง
  - start ขนาด 1 bit เมื่อได้ 1 ให้เริ่มทำงานตาม pRAM
  - result ขนาด 1 bit เมื่อได้ 1 ให้เริ่มส่งคำตอบที่อยู่ใน rRAM
  - clk ขนาด 1 bit เป็นสัญญาณ clock สำหรับใช้ในการให้จังหวะ
- ส่วนนี้สำหรับเป็นตัวช่วยในการทำงานของวงจร ในการทดสอบ grader จะไม่ได้ส่งส่วนนี้ให้โดยตรง
- [pRAM ขนาด 256 x 14 bits] สำหรับเก็บโปรแกรม ที่ได้รับเข้ามา สามารถเลือกเป็นแบบใดก็ได้

### วงจรมี output คือ

- valid ขนาด 1 bit เป็น 1 เมื่อทำงานตามโปรแกรมเสร็จเรียบร้อยพร้อมส่งคำตอบ
  - done ขนาด 1 bit เป็น 1 เมื่อการทำงานเสร็จสิ้นและส่งผลคำตอบเรียบร้อย หรือตอนเริ่มต้น ครั้งแรกที่พร้อมรับคำสั่ง reset / progLoad / start
  - output ขนาด 8 bits เป็นค่า output ที่ได้จากการทำงาน โดยแสดงผลลัพธ์ที่อยู่ใน rRAM เป็น เลขฐาน 16 โดยจะแสดงค่าห้องจากไดร์บสัญญาณ result
- ส่วนนี้สำหรับเป็นตัวช่วยในการทำงานของวงจร ในการทดสอบ grader จะไม่ได้อ่านส่วนนี้โดยตรง
- [7-segment 2 ตัว] แสดงค่าผลลัพธ์ของ output สำหรับให้ตรวจสอบผลลัพธ์
  - [rRAM ขนาด 256 x 8 bits] สำหรับเก็บคำตอบ สามารถเลือกเป็นแบบใดก็ได้

### ข้อมูลเพิ่มเติมสำหรับการพัฒนา CPU

- สามารถใช้อุปกรณ์ อะไรก็ได้เพิ่มเติมที่มีอยู่ใน Digital แบบมาตรฐาน
- จำนวนบรรทัดของคำสั่งจะอยู่ในช่วง 1 - 256 คำสั่ง
- คำสั่งโปรแกรมที่ส่งมาให้จัดเก็บเข้า pRAM โดยเมื่อทำงานให้อ่านคำสั่งของโปรแกรมตามลำดับ โดยให้ทำงานตั้งแต่บรรทัดแรกไปจนเจอคำสั่งที่ให้หยุดการทำงาน
- CPU นี้จะใช้การคำนวนแบบ 8 bits เป็นหลัก
- CPU นี้จะต้องออกแบบให้เป็น Multiple Cycle CPU หรือ Pipeline ไม่อนุญาตให้ทำเป็น Single Cycle CPU
- การ implement ส่วนของ control unit จะทำเป็น sequential logics หรือใช้ microprogram ก็ได้
- ระบบตัวเลขสำหรับคำนวนทางคณิตศาสตร์จะเป็นแบบ 2's complement ยกเว้นจะระบุเป็นอย่างอื่น
- แต่ละคำสั่ง จะออกแบบให้ใช้จำนวน clock เท่ากันหรือไม่เท่ากันก็ได้
- ต้องมีการออกแบบ register เพื่อเก็บค่าสถานะหรือข้อมูลต่างๆเพิ่มเติมเอง ตามความเหมาะสม
- "ไม่จำเป็นต้องทำได้ครบถ้วนคำสั่ง บาง test case อาจจะใช้เพียงไม่กี่คำสั่งก็ได้"
- กำหนดให้ใช้ positive edge clock ในการ Synchronous วงจร

แต่ละคำสั่งจะมีโครงสร้างคำสั่งดังนี้

13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	---	---	---	---	---	---	---	---	---	---

Opcode 6 bits	Operand 8 bits
---------------	----------------

โครงสร้างของ instruction ขนาด 14 bits แบ่งเป็น opcode 6 bits และ operand 8 bits

Opcode และค่าอธิบายเป็นไปดังตารางด้านล่าง

Opcode	คำสั่ง	ความหมาย
000000	NOPE	ไม่ต้องทำงานอะไร ข้ามไปทำงานคำสั่งถัดไป
000001	accA ← Operand	นำค่า 8 bits ของ operand ไปเก็บไว้ใน accA
000010	accB ← Operand	นำค่า 8 bits ของ operand ไปเก็บไว้ใน accB
000011	accA ← accB	นำค่าจาก accB ไปเก็บไว้ใน accA
000100	accB ← accA	นำค่าจาก accA ไปเก็บไว้ใน accB
000101	regC ← accA	นำค่าจาก accA ไปเก็บไว้ใน regC
000110	accA ← regC	นำค่าจาก regC ไปเก็บไว้ใน accA
000111	regD ← accA	นำค่าจาก accA ไปเก็บไว้ใน regD
001000	accA ← regD	นำค่าจาก regD ไปเก็บไว้ใน accA
001001	regC ← M	นำค่าจาก input M ไปเก็บไว้ใน regC
001010	regC ← N	นำค่าจาก input N ไปเก็บไว้ใน regC
001011	regD ← M	นำค่าจาก input M ไปเก็บไว้ใน regD
001100	regD ← N	นำค่าจาก input N ไปเก็บไว้ใน regD
001101	regC ← M regD ← N	นำค่าจาก input M ไปเก็บไว้ใน regC นำค่าจาก input N ไปเก็บไว้ใน regD
001110	accA ← regC accB ← regD	นำค่าจาก regC ไปเก็บไว้ใน accA นำค่าจาก regD ไปเก็บไว้ใน accB
001111	accA ← pRAM_adr[Operand]	นำค่าจาก pRAM ใน address ที่ระบุโดย operand ไปเก็บไว้ใน accA โดยจัดเก็บเฉพาะ 8 bits ด้านขวาเท่านั้น
010000	accA ← rRAM_adr[Operand]	นำค่าจาก rRAM ใน address ที่ระบุโดย operand ไปเก็บไว้ใน accA
010001	rRAM_adr[Operand] ← accA	นำค่าจาก accA ไปเก็บไว้ใน rRAM ใน address ที่ระบุโดย operand
010010	Jump to address	ข้ามไปทำคำสั่งใน address ตามที่ระบุใน Operand แบบไม่มี

Opcode	คำสั่ง	ความหมาย
	Operand	เงื่อนไข
010011	Jump to address Operand if eq	ข้ามไปท่าคำสั่งใน address ตามที่ระบุใน Operand ถ้า equal flag เป็น 1
010100	Jump to address Operand if gr	ข้ามไปท่าคำสั่งใน address ตามที่ระบุใน Operand ถ้า greater flag เป็น 1
010101	Jump to address Operand if le	ข้ามไปท่าคำสั่งใน address ตามที่ระบุใน Operand ถ้า lesser flag เป็น 1
010110	Jump to address Operand if eq or gr	ข้ามไปท่าคำสั่งใน address ตามที่ระบุใน Operand ถ้า equal flag หรือ greater flag เป็น 1
010111	Jump to address Operand if eq or le	ข้ามไปท่าคำสั่งใน address ตามที่ระบุใน Operand ถ้า equal flag หรือ lesser flag เป็น 1
100000	$accA \leftarrow accA + accB$	นำค่าจาก accA มาบวกกับ accB และไปเก็บที่ accA (ไม่ต้องสนใจกรณีผลบวกเกินขอบเขต) คำนวณแบบ 2's complement
100001	$accA \leftarrow accA - accB$	นำค่าจาก accA มาลบกับ accB และไปเก็บที่ accA (ไม่ต้องสนใจกรณีผลลบเกินขอบเขต) คำนวณแบบ 2's complement
100010	$accA \leftarrow accA * accB$	นำค่าจาก accA[3..0] มาคูณกับ accB[3..0] และไปเก็บที่ accA (คิด 4 bits ดังนั้นจะมองเป็นเลขบวกอย่างเดียว)
100011	$accA \leftarrow accA / accB$	นำค่าจาก accA คิดแบบ binary ไม่ดู signed bit มาหารกับ accB และไปเก็บที่ accA
100100	$accA \leftarrow accA \% accB$	นำค่าจาก accA คิดแบบ binary ไม่ดู signed bit มา mod กับ accB และไปเก็บที่ accA
100101	$accA \leftarrow accA ^ accB$	นำค่าจาก accA[2..0] ยกกำลัง accB[2..0] และไปเก็บที่ accA (ไม่ต้องสนใจกรณีผลลัพธ์เกินขอบเขต)
101000	$accA \leftarrow NOT(accA)$	กลับบิตของ accA และเก็บไว้ที่ accA
101001	$accA \leftarrow accA AND accB$	นำค่าจาก accA มา bitwise AND กับ accB และไปเก็บที่ accA
101010	$accA \leftarrow accA OR accB$	นำค่าจาก accA มา bitwise OR กับ accB และไปเก็บที่ accA
101011	$accA \leftarrow accA XOR accB$	นำค่าจาก accA มา bitwise XOR กับ accB และไปเก็บที่ accA
101100	$accA \leftarrow accA << accB$	นำค่าจาก accA มา logical shift left ตามค่าของ accB[2..0] และไปเก็บที่ accA โดย shift left ไม่เกิน 7 bits

Opcode	คำสั่ง	ความหมาย
101101	$accA \leftarrow accA \lll accB$	นำค่าจาก accA มา rotate logical shift left ตามค่าของ accB[2..0] และไปเก็บที่ accA โดย shift left ไม่เกิน 7 bits
101110	$accA \leftarrow accA \ggg accB$	นำค่าจาก accA มา logical shift right ตามค่าของ accB[2..0] และไปเก็บที่ accA โดย shift right ไม่เกิน 7 bits
101111	$accA \leftarrow accA \ggg accB$	นำค่าจาก accA มา rotate logical shift right ตามค่าของ accB[2..0] และไปเก็บที่ accA โดย shift right ไม่เกิน 7 bits
110000	accA CMP accB	เทียบค่า accA กับ accB คำนวณแบบ 2's complement - ถ้า $accA == accB$ ค่า equal flag จะเป็น 1 - ถ้า $accA > accB$ ค่า greater flag จะเป็น 1 - ถ้า $accA < accB$ ค่า lesser flag จะเป็น 1
110001	isPrime(accA)	ตรวจสอบว่า accA เป็นจำนวนเฉพาะหรือไม่ ถ้า - accA เป็นจำนวนเฉพาะ ค่า equal flag จะเป็น 1 - กรณีนี้ไม่กระทบกับ greater flag และ lesses flag ให้คิด accA และ accB แบบเลขฐาน 2 ปกติ ไม่มีเลขลบ
110010	accB   accA	ทดสอบว่า accA ที่เป็นตัวตั้งหารด้วย accB ลงตัวหรือไม่ - ถ้าลงตัวให้ equal flag เป็น 1 - กรณีนี้ไม่กระทบกับ greater flag และ lesses flag ให้คิด accA และ accB แบบเลขฐาน 2 ปกติ ไม่มีเลขลบ เช่น $accA = 20$ และ $accB = 5$ จะได้ว่า $5 20$ คือ 20 หารด้วย 5 ลงตัว จะทำให้ equal flag เป็น 1
110011	$rRAM[0x0E:0x0F] \leftarrow LCM(rRAM\_adr[Operand[7:4]], rRAM\_adr[Operand[3:0]])$	คำนวณหาค่าคูณร่วมน้อย LCM(m,n) โดย m คือค่าใน rRAM ตำแหน่งตาม operand[7:4] n คือค่าใน rRAM ตำแหน่งตาม operand[3:0] ให้คิดตัวเลขแบบ binary เป็นจำนวนเต็มบวกเท่านั้น  นำผลลัพธ์ที่ได้เก็บไว้ที่ rRAM[0x0E] และ rRAM[0x0F] โดยค่า most significant บิต result[15:8] เก็บที่ rRAM[0x0E] โดยค่า least significant บิต result[7:0] เก็บที่ rRAM[0x0F]  <a href="https://en.wikipedia.org/wiki/Least_common_multiple">https://en.wikipedia.org/wiki/Least_common_multiple</a>
110100	$rRAM[0x0A:0x0B] \leftarrow FAC(accA[2:0])$	คำนวณหา Factorial(n) โดย n คือค่าใน accA เฉพาะบิต [2:0] ดังนั้นค่าของ n จะอยู่ระหว่าง 0 - 7 เท่านั้น  นำผลลัพธ์ที่ได้เก็บไว้ที่ rRAM[0x0A] และ rRAM[0x0B] โดยค่า most significant บิต result[15:8] เก็บที่ rRAM[0x0A] โดยค่า least significant บิต result[7:0] เก็บที่ rRAM[0x0B]  ในคำสั่งนี้ให้ใช้ การวนลูปผ่าน ASM ไม่อนุญาตให้ใช้ ROM  <a href="https://en.wikipedia.org/wiki/Factorial">https://en.wikipedia.org/wiki/Factorial</a>

Opcode	คำสั่ง	ความหมาย
110101	$rRAM[0x09] \leftarrow \max(pRAM[accA:accB])$	<p>ค่านวนหาค่าที่มากที่สุด ของค่าใน pRAM[7:0] (คิดเฉพาะ 8 มิต ด้านขวาเท่านั้น) ต่อหนึ่งที่ระบุโดย accA ถึง accB เช่น ถ้า accA เป็น 10 และ accB เป็น 20 จะหมายถึงหาค่าที่มากที่สุดจาก pRAM[0x0A : 0x14] นำค่าตอบ ที่ได้ไปเก็บที่ rRAM[0x09]</p> <p>ให้ใช้การเปรียบเทียบแบบ 2's complement</p> <p>ในการนี้โจทย์จะตั้งให้ <math>accA \leq accB</math> เชื่อ</p>
111111	STOP	หยุดการทำงาน ไม่ต้องทำคำสั่งถัดไป และรอสัญญาณ result

accA, accB, regC, regD คือ Accumulator A, Accumulator B, Register C, Register D ตามลำดับ โดยมีขนาด 8 bits

### การทำงานในทีม

- ให้ทำงานเป็นกลุ่ม กลุ่มละ 3-4 คน โดยนิสิตสามารถจับกลุ่มกันเองได้
  - กลุ่ม 2 คนก็ได้ พอนุ่ม แต่ถ้ามีเพื่อนไม่มีกลุ่ม ก็อาจจะขอให้เพื่อนร่วมกลุ่มด้วย
- นิสิตสามารถปรึกษากันระหว่างกลุ่มได้ แต่ห้ามคัดลอก
- สามารถใช้ generative AI ในการช่วยวิเคราะห์และพัฒนาวางแผนจาร์ได้

### การให้คะแนน (draft)

- เอกสารด้านเทคนิคไม่เกิน 10 หน้า (10 คะแนน)
  - แนวคิดการออกแบบ CPU
  - ASM Chart หรือ FSM Chart
  - การออกแบบและพัฒนาส่วน Data Path
  - การออกแบบและพัฒนาส่วน Control Unit
- ส่วนการนำเสนอ (10 คะแนน)
  - สุ่มตัวแทนที่จะมานำเสนอ และแสดงว่าทุกคนควรรู้เรื่องและอธิบายได้
  - ให้จัดทำสไลด์ประกอบการนำเสนอ
  - ระยะเวลานำเสนอ 5-10 นาที
- การทำงานตาม test case (80 คะแนน)
  - ในแต่ละ โปรแกรมทดสอบจะมีความแตกต่างกันทั้งเรื่องจำนวนคำสั่งที่ใช้ บรรทัดของคำสั่ง
  - สำหรับโปรแกรมทดสอบเดียวกัน ก็จะมีหลาย test case เพื่อทดสอบสิ่งสัญญาณ input เช่น M, N, reset, start, result ในเวลาที่ไม่เหมือนกันเพื่อตัดการตอบสนองของวงจรที่สร้างขึ้น
- ทุกๆ สมาชิกที่เกินกว่า 4 คน จะถูกหักคะแนนละ 5 คะแนน (-5 คะแนน)

### Extra Credit

- กลุ่ม (จำนวนไม่เกิน 5 คน) ที่สามารถส่ง wangjgrader ได้ถูกต้องทุก test case จำนวน 5 กลุ่ม แรก (ในกลุ่มต้องทำเอง ห้ามทูลวิธีเอาของกลุ่มอื่นมาดัดแปลงแล้วส่ง) จะได้รางวัลพิเศษ (ต้องแสดงรหัสและชื่อสมาชิกในไฟล์ .dig และแสดงหน้าที่ของแต่ละคน โดยยึดรายชื่อตามไฟล์ครั้งแรก ที่ส่งผ่านครบทุก test case
  - ถือว่าสอบผ่านได้ S หังกลุ่ม โดยไม่มีพิจารณาค่าคะแนนแล้วปะและคะแนนสอบ
  - คนที่เป็นหลักในการออกแบบและพัฒนาจำนวนไม่เกิน 2 คนต่อกลุ่ม จะได้ S\* คือใน Certificate ยอดเยี่ยมประจำวิชา

## ตัวอย่างการทำงาน

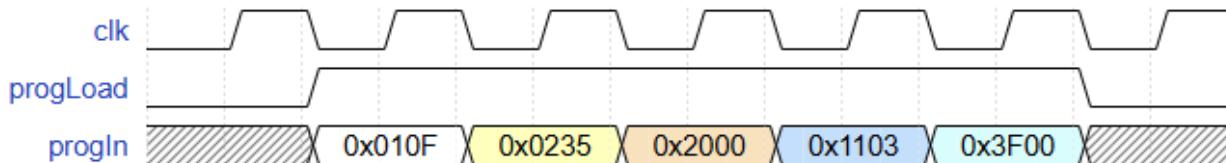
กำหนดให้ข้อมูลโปรแกรมทดสอบเป็นดังนี้

0b 00 0001 0000 1111	accA $\leftarrow$ Operand
0b 00 0010 0011 0101	accB $\leftarrow$ Operand
0b 10 0000 0000 0000	accA $\leftarrow$ accA + accB
0b 01 0001 0000 0011	rRAM_adr[Operand] $\leftarrow$ accA
0b 11 1111 0000 0000	STOP

เริ่มต้นจะมีสัญญาณ reset เพื่อให้วงจรทำการตั้งค่าของระบบ โดยจะมี clock อย่างน้อยจำนวน 20 clock ก่อนจะมีสัญญาณ progLoad

รอสัญญาณ progLoad เป็น 1 จะให้เริ่มรับคำสั่งทีละคำสั่งผ่านทาง progIn ไปเก็บไว้ใน pRAM โดยเริ่มต้นที่ address 0x00 และค่อยๆเพิ่ม address ทีละ 1 จนกว่าสัญญาณ progLoad เป็น 0 ซึ่งหมายถึง program ได้ถูกส่งให้ครบถ้วนแล้ว

ในการนี้จะส่งมาทั้งหมด 5 คำสั่ง ให้จัดเก็บที่ pRAM address 0x00 - 0x04



รอสัญญาณ start เป็น 1 (จะเป็น 1 สักๆ) จะให้เริ่มทำการคำสั่งตามลำดับใน pRAM

เริ่มจากให้ accA  $\leftarrow$  15

accB  $\leftarrow$  53

accA  $\leftarrow$  15 + 53 = 68

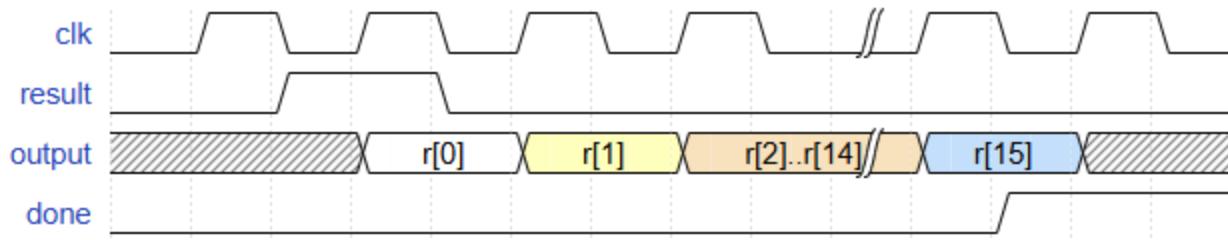
rRAM\_adr[3]  $\leftarrow$  68

เมื่อทำงานเสร็จ ให้กำหนดสัญญาณ valid = 1

จากนั้นรอลุนได้สัญญาณ result = 1 จึงเริ่มส่งค่าของ rRAM ตั้งแต่ address 0x00 ไปจนถึง 0x0F ซึ่งจะได้

rRAM[0x00]	$\rightarrow$ 0
rRAM[0x01]	$\rightarrow$ 0
rRAM[0x02]	$\rightarrow$ 0
rRAM[0x03]	$\rightarrow$ 68 $\rightarrow$ 0x44
...	
rRAM[0x0F]	$\rightarrow$ 0

เมื่อแสดงผลครบ ให้ส่งสัญญาณ done ออกเป็น 1 เพื่อรอสัญญาณการทำงานใหม่ โดยต้องสามารถรับสัญญาณ reset, progLoad และ start ได้อย่างต่อเนื่อง



## Template

Template สามารถเข้าถึงได้จาก [TEMPLATE Project 2568.dig](#)

## Testcase

ไฟล์ที่ไว้สร้าง test case สามารถทดลองใช้ได้จาก excel (ให้โหลดลงที่เครื่องมาทดลอง ตัว testcase จะอยู่ที่คอลัมน์ P)

1	Program	Test1	simulation												Meaning	Assembly	testcase format											
			M	N	progIN	reset	progLoad	start	result	clk	valid	done	output	M		N	Machine code	M	N	progIN	reset	progLoad	start	result	clk	valid	done	output
45	0	0	271	0	1	0	0	0	0	0	0	0	0	accA += Operand	00000100001111	0	0	271	0	1	0	0	0	0	0	0	x	
46	0	0	271	0	1	0	0	1	0	0	0	0	0	inc accA by 1 (pRAM[0x00])	00000100001111	0	0	271	0	1	0	0	1	0	0	0	x	
47	0	0	565	0	1	0	0	0	0	0	0	0	0	accA += Operand	00000100110101	0	0	565	0	1	0	0	0	0	0	0	x	
48	0	0	565	0	1	0	0	1	0	0	0	0	0	inc accA by 2 (pRAM[0x01])	00000100110101	0	0	565	0	1	0	0	1	0	0	0	x	
49	0	0	8192	0	1	0	0	0	0	0	0	0	0	accA + accB	10000000000000	0	0	8192	0	1	0	0	0	0	0	0	x	
50	0	0	8192	0	1	0	0	0	1	0	0	0	0	inc accA by 3 (pRAM[0x02])	10000000000000	0	0	8192	0	1	0	0	1	0	0	0	x	
51	0	0	4355	0	1	0	0	0	0	0	0	0	0	RAM_add[Operand] = accA	0100010000001111	0	0	4355	0	1	0	0	0	0	0	0	x	
52	0	0	4355	0	1	0	0	0	1	0	0	0	0	inc accA by 4 (pRAM[0x03])	0100010000001111	0	0	4355	0	1	0	0	1	0	0	0	x	
53	0	0	16128	0	1	0	0	0	0	0	0	0	0	STOP	11111100000000	0	0	16128	0	1	0	0	0	0	0	0	x	
54	0	0	16128	0	1	0	0	1	0	0	0	0	0	inc accA by 5 (pRAM[0x04])	11111100000000	0	0	16128	0	1	0	0	1	0	0	0	x	
55	0	0	0	0	0	0	1	0	0	0	0	0	0	inc accA by 6 (pRAM[0x05])	11111100000000	0	0	0	0	0	1	0	0	0	0	0	x	
56	0	0	0	0	0	0	1	0	1	0	0	0	0	inc accA by 7 (pRAM[0x06])	11111100000000	0	0	0	0	0	1	0	1	0	0	0	x	
57	0	0	0	0	0	0	0	0	0	0	0	0	0	inc accA by 8 (pRAM[0x07])	11111100000000	0	0	0	0	0	0	0	0	0	0	0	x	
58	0	0	0	0	0	0	0	0	1	0	0	0	0	inc accA by 9 (pRAM[0x08])	11111100000000	0	0	0	0	0	0	0	1	0	0	0	x	

[Testcase template 2568.xlsx](#)

## Test case นะครับ

- Test case ใน grader ที่ให้ทดสอบจะสอดคล้องกับ test case ในไฟล์ [Testcase template 2568.xlsx](#)
- 01\_P01T01 ทดสอบการโหลดโปรแกรมและการส่งค่าทั่วไป
- 02\_P01T02 request result เป็นครั้งที่ 2 หลังจากส่ง output ออกมานแล้ว
- 03\_P02T01 Test simple jump
- 04\_P03T01 pRamLoad\_Test
- 05\_P03T02 pRamLoad\_Test
- 06\_P04T01 isPrime\_Test
- 07\_P05T01 isPrime + CMP
- 08\_P06T01 ทดสอบการอ่านค่า M และ N
- 09\_P06T02 ทดสอบการอ่านค่า M และ N / แล้วก็ทดสอบการ run ใหม่โดยไม่ได้โหลดโปรแกรมใหม่
- 10\_P07T01 ทดสอบคำสั่งพิเศษ LCM
- 11\_P07T02 ทดสอบคำสั่งพิเศษ LCM โดยให้ค่าคำตอบจะเกิน 8 บิต และเก็บตัวตนไว้ที่ address 0xF ซึ่งจะໄວ่เก็บตำแหน่งด้วย
- 12\_P08T01 ทดสอบการ + - \* / % ^
- 13\_P08T02 ทดสอบการ + - \* / % ^ โดยการสั่งแก้ไขบางส่วนของโปรแกรมแล้วทำงานเลย
- 14\_P09T01 ทดสอบ bitwise operation NOT AND OR XOR <<
- 15\_P09T02 ทดสอบ bitwise operation NOT AND OR XOR << แล้วก็ reset ระหว่างทำงาน
- 16\_P01T03 ทดสอบการโหลดโปรแกรมและการส่งค่าทั่วไป และทดสอบ request result ข้ามตัว
- 17\_P02T02 Jump ทุกรูปแบบ และมีการหยุดระหว่างทาง
- 18\_P03T03 pRamLoad\_Test extend
- 19\_P04T02 isPrime\_Test Extend
- 20\_P04T03 accB | aacA ทดสอบว่า accA หารด้วย accB ลงตัวหรือไม่

- 21\_P05T02 isPrime\_Test + flag condition
- 22\_P06T03 ทดสอบการอ่านค่า M และ N ใส่ regC regD
- 23\_P07T03 ทดสอบคำสั่งพิเศษ LCM
- 24\_P08T03 ทดสอบการ + - \* / % ^ โดยการสั่งแก็บงานส่วนของโปรแกรมแล้วทำงานเลย
- 25\_P09T03 ทดสอบ bitwise operation NOT AND OR XOR << <<< และ reset ระหว่างทำงาน
- 26\_P10T01 ทดสอบคำสั่งพิเศษ factorial
- 27\_P10T02 ทดสอบคำสั่งพิเศษ factorial
- 28\_P11T01 ทดสอบคำสั่งพิเศษ max
- 29\_P11T02 ทดสอบคำสั่งพิเศษ max

## กำหนดการและแนวทางในการส่ง

### วันประกาศโครงการ

- ให้นิสิตจับกลุ่มและแจ้งรายละเอียดกลุ่ม โดยให้สมาชิกเพียงคนเดียวเป็นตัวแทนในการกรอก (รวมถึงต้องใช้ account นี้ในการ update และส่งข้อมูลตลอด)
- ให้กรอกชื่อทีมและสมาชิกภายใน วันจันทร์ที่ 20 ต.ค. 68 เวลา 14:00
- เริ่มออกแบบและพัฒนาเว็บ

### ก่อนวันกำหนดส่งรายหนึ่งภาคในวันที่ 27 ต.ค. 68

- แจ้งตัวอย่าง test case ที่ใช้ในการทดสอบเบื้องต้น (ชึ้นนิสิตจะทราบค่าสั่งที่มี จำนวนคำสั่ง จำนวน clock มากที่สุดที่ยอมให้ใช้)
- เปิด grader ให้ทดลองส่งบาง test case

### ก่อนวันกำหนดส่ง 29 ต.ค. 68 06:00

- ทยอยเพิ่มและแจ้ง Test case (เกือบ) ทั้งหมดที่ใช้ในการทดสอบ
- ให้จดจำลักษณะการนำเสนอ

### ก่อนกำหนดส่งเล็กน้อย 29 ต.ค. 68

เวลาประมาณ 19:00

- เพิ่ม Test case ทั้งหมดที่ใช้ในการทดสอบ
- เริ่มนับเวลาส่งอย่างเป็นทางการเพื่อคัดเลือกทีมที่ส่งได้ครบทุก test case ตาม Extra Credit

### ช่วงการนำเสนอ

- ให้ Upload เอกสารสำหรับนำเสนอให้เสร็จ
- นำเสนอคร่าวๆ ประมาณ 5-10 นาที
- โดยคนนำเสนอและตอบคำถามจะถูกสุ่มจากสมาชิกคนใดคนหนึ่ง
- นำเสนอทีละกลุ่ม ตามลำดับการจดจำ

### หลังการนำเสนอ

- มีเวลาให้ทำการพัฒนา CPU เพิ่มเติม
- Grader จะปิดรับการส่งในวันที่ 30 ตุลาคม 2568 ตอนเที่ยงคืน
- ให้ Upload เอกสารต่างๆ เช่นไฟล์สำหรับนำเสนอ รายงาน ไฟล์โปรแกรม และ capture หน้าจอ คะแนน grader ผ่านทาง MCV ให้เสร็จก่อนเที่ยงคืน

### หลังวันกำหนดส่ง 1-2 พฤศจิกายน 2568

- พักผ่อนหรือเตรียมตัวสำหรับวิชาต่อไป
- ขอให้นิสิตทุกคนโชคดีและสนุกกับการเรียน

มีความต้องการเพิ่มเติมสามารถ เพิ่ม comment มาได้เลยนะครับ

1. เวลา reset และค่าใน register กับ flags จะเปลี่ยนใหม่ครับ
  - ควร reset ให้เรียบร้อยครับ โดยค่า default ควรจะเป็น 0
2. การสุ่มตัวแทนสุ่มจำนวนกี่คนจะ เป็นสลับกันตอบคำถามมั้ยจะ และรูปเล่นมีตัวอย่างให้มั้ยครับ
  - สุ่ม 1 คน ในกลุ่ม ทั้งนำเสนอและตอบคำถาม คนเดียวครับ
  - เล่นไม่มีตัวอย่างครับ
3. Data Input 14 bit ชื่อว่า progIn หรือ progIN จะครับ พอดีว่าในเอกสารกับใน test case ไม่ตรงกันครับ (ผมใช้ progIN และส่งใน grader ผ่านนะครับ น่าจะ progIN แหล่ง)
  - progIn ครับ ผมแก้ไขใน grader template และ excel และ
4. สามารถใช้ single cycle processor ในการออกแบบได้หรือเปล่าครับ
  - ไม่ได้ครับ
5. reset ต้องเคลียร์ทุกบิตใน rRam เป็น 0 ทุกช่องเลยหรือครับ หรือแค่ 20 บิตแรกก็พอครับ
  - ควรจะ clear ครับ โดยเฉพาะ 16 bytes แรก
6. สามารถใช้อุปกรณ์ อะไหล่ได้เพิ่มเติมที่มีอยู่ใน Digital แบบมาตรฐาน:  
อันนี้ ROM ใช้ได้ใช้ใหม่ครับ ผมจำได้ว่าอาจารย์ไม่ให้ใช้ ใน lab/สอบ แต่ว่าครั้งนี้ (project) สามารถใช้ได้ใช้ใหม่ครับ
  - ใช้ได้ครับ รอบนี้เลือกได้เลย
7. ASM/FSM/Datapath จะเป็นต้องใช้โปรแกรมวัดใหม่ครับ หรือว่ามือได้
  - วัดมือได้ครับ
8. progLoad จะเป็น 1 ตั้งแต่ตอนที่ clock เป็น 0 ตลอดรีเปล่าครับ หรือว่าสามารถเกิดเหตุการณ์ที่ progLoad เริ่มเป็น 1 ตอนที่ clock เป็น 1 พอดีได้
  - จะเป็น 1 ตอนที่ clock เป็น 0 เพื่อรอไว้ก่อนครับ
9. การส่ง 5 กลุ่มแรกต้องส่งที่ไหนครับผม ต้องส่งอะไรบ้าง และจะเผยแพร่ test case อีกทีกี่ โงนั้นครับ
  - ตัวเต็มก็ส่งผ่าน grader ครับ แต่ตอนนี้ยังไม่ให้เห็นนะ
  - จะปล่อย test case เรื่อยๆ ครับ
  - คาดว่าจะปล่อยตัวเต็มในวันพุธเวลา 18:00
10. มีวิธีใส่ค่าลง RAM พร้อมกันหลาย address มั้ยครับ
  - ไม่แน่ใจนะ ถ้าใครทราบมาแชร์หน่อย
11. คะแนนส่วนการทำงานตาม test case (80 คะแนน) นี่ดูจาก grader อย่างเดียวรึป่าวครับ
  - ใช้ครับ แต่จำนวน test case ยังไม่เพิ่มเรื่อยๆ เท่าที่เปิดคือเปิด แต่ตัวเต็มจะมีมากกว่านี้

12. Test Case ใน Grader ตรงกับใน Template หรือไม่ครับ เนื่องจากเห็นว่า Test Case ที่ให้มามีข้อผิดพลาดบางส่วน จึงอยากรบราบว่าใน Grader เป็นแบบเดียวกันหรือไม่ หากแตกต่างจะทราบได้อย่างไรว่าไม่เป็นข้อผิดพลาดที่ Test Case แต่หากเหมือนกันก็ได้ครับ

- เมื่อ้อนกันครับ
- ช่วงนี้ทดสอบ test case ต่างๆ ถ้าเจอตรงไหนไม่ถูกต้อง ก็แจ้งมาได้ครับ

13. ทำไมไม่การประการเลยคะว่ามีเทสเคสใหม่ เพื่อนๆ ได้ไปว่ามีมาใหม่ 😊😊

- กำลังเพิ่ม testcase และก็ทดสอบไป เลยยังไม่ได้แจ้งทางการ ขออภัยด้วยนะครับ
- ที่เพื่อนๆ ทราบน่าจะ เพราะว่า ส่วนจะผ่าน grader เลยเห็นว่ามีการเพิ่ม test case อุ้ย

14. หลังจากทำงานเสร็จ (done หลังการทำงานครั้งแรกแล้ว) ต้อง reset register และ flag ใหม่ครับ

- ไม่ต้องครับ
- รอคำสั่ง control line ถัดๆ ไป

15. จะมี testcase ที่ result หลังจาก ทำงานเสร็จ เลยใหม่ครับ เช่น หลังจาก opcode = 111111 1 cycle และ result = 1 เลย

- ไม่มีครับ

กลุ่มที่หนึ่ง ส่งแล้วถูกต้องทั้งหมด ให้มาโพสต์ด้านล่างนี้เลยนะครับ ໄลไป 10 อันดับ (ที่ส่งผ่าน จะมีการตรวจ  
อีกครั้งว่า เป็นไปตามข้อกำหนดหรือไม่)

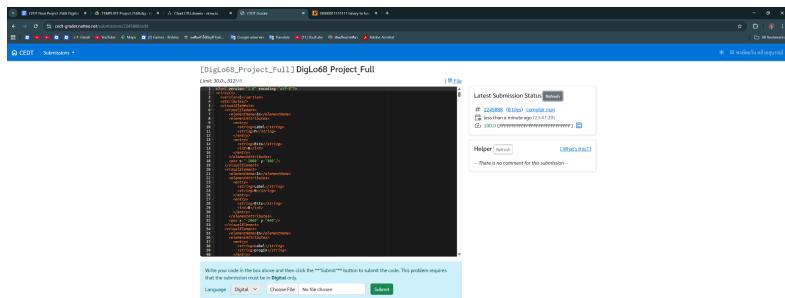
- แสดงชื่อกลุ่ม
- Capture หน้าจอที่ถูกต้องทุกข้อ

ผ่านหมวดอันดับ 1

## R2D2 RAM

RRDD Random Access Memory

- Norraphat Rathasamuth
- Paweenarin Jearanilkulchai
- Pongtawan Klysubun
- Pathavee Siripathavanich



</list>

ผ่านหมวดอันดับ 2

Only "U"

Only "U"

- Tanapat Panakorn
- Noppawit Athirattrueangkit
- Yarnnaphart Chamnongsooth
- Ittipong Kornpongkan

The screenshot shows a web browser window with the URL <https://cedt-grader.rattree.net/submissions/2247823/edit>. The page title is "CEDT Submissions". The main content area displays the code for "DigLo68\_Project\_Full" with line numbers 1 through 40. To the right of the code editor, there is a "Latest Submission Status" box showing "# 2247823 (1) tries compiler.map 2 minutes ago (06:30:20) 100.0 [XXXXXXXXXXXXXXXXXXXXXXXXXXXX]". Below this is a "Helper" section with a "Refresh" button and a link "(What's this?)". A note at the bottom says "With your code in the box above and then click the \*\*\*Submit\*\*\* button to submit the code. This problem runs once." The footer of the page includes the copyright notice "© 2025 cedt-grader | Revision: 4.1.3cu".

## ผ่านหมวดอันดับ 3

### Submission: #2247846

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <circuit>
3   <version>2</version>
4   <attributes/>
5   <visualElements>
6     <visualElement>
7       <elementName>In</elementName>
8       <elementAttributes>
9         <stringLabel><string>
10        <string>N</string>
11      </entry>
12      <entry>
13        <stringBits><string>
14          <int>8</int>
15        </string>
16      </entry>
17      <elementAttributes>
18        <pos x="860" y="-2320"/>
19      </visualElement>
20      <visualElement>
21        <elementName>In</elementName>
22        <elementAttributes>
23          <entry>
24            <stringLabel><string>
25              <string>W</string>
26            </string>
27          </entry>
28          <stringBits><string>
29            <int>8</int>
30          </string>
31        </elementAttributes>
32        <pos x="860" y="-2260"/>
33      </visualElement>
34      <visualElement>
35        <elementName>In</elementName>
36        <elementAttributes>
37          <entry>
38            <stringLabel><string>
```

### Submission Detail

User	<a href="#">6833262521</a> สิทธิโชค ภูร่อง
Problem	<a href="#">DigLo68 Project Full</a> <a href="#">DigLo68_Project_Full</a>
	<a href="#">Edit</a>   <a href="#">Stat</a>
Tries	15
Language	Digital
Submitted	about 2 hours ago (at October 30, 2025 06:42)
Graded	about 2 hours ago (at October 30, 2025 06:42)
Points	100.0/100
Result	PPPPPPPPPPPPPPPPPPPPPPPPPPPP
Runtime	7.83 s
Memory	158220 kb
Compiler result	<a href="#">View</a>
Grading Task Status	done

## ผ่านหมวดอันดับ 4

JohnnyLoveDiglo

6833244221 Waranthon Wanichnam

6833169921 Punnaphat Siangchin (Main Circuit Developer)

6833167621 Pitichod Jitkaew (Main Circuit Developer)

6833146421 Benyapha Sa-Ardmuang

CEDT Submissions ▾

[DigLo68\_Project\_Full] DigLo68\_Project\_Full

Limit: 30.0s, 511MB

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <circuit>
3 <version>2</version>
4 <elementName>
5 <visualElement>
6   <elementName>In</elementName>
7   <elementAttributes>
8     <entry>
9       <stringLabel>/string<
10      <stringProgIn>/string<
11      <int>14</int>
12    </entry>
13    <entry>
14      <stringBits>/string<
15      <int>14</int>
16    </entry>
17  </elementAttributes>
18  <visualElement>
19    <elementName>In</elementName>
20    <elementAttributes>
21      <entry>
22        <stringLabel>/string<
23        <stringProgIn>/string<
24        <int>14</int>
25      </entry>
26      <entry>
27        <stringBits>/string<
28        <int>14</int>
29      </entry>
30    </elementAttributes>
31    <pos x="340" y="380"/>
32  </visualElement>

```

Latest Submission Status Refresh

# 2248034 (15 tries) compiler msg  
less than a minute ago (08:44:23)  
100.0 [PPPPPPP...]

Helper Refresh [What's this?]

-- There is no comment for this submission --

© 2025 cafe-grader | Revision: 4.1.3cu

## ผ่านหมวดอันดับ 5

### TK Fast Work

- 6833102821 ຮັນກຸດ ແກ້ວຫາວັງໝັ້ນ
- 6833023921 ຄົນພັດນີ້ ເລີ່ມເກີຍຮົດສຸກລ
- 6833064621 ພະຈູກ ແຊ້ໂອ້ວ

CEDT Submissions ▾

[DigLo68\_Project\_Full] DigLo68\_Project\_Full

Limit: 30.0s, 511MB

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <circuit>
3 <version>2</version>
4 <elementName>
5 <visualElement>
6   <elementName>In</elementName>
7   <elementAttributes>
8     <entry>
9       <stringLabel>/string<
10      <stringProgIn>/string<
11      <int>14</int>
12    </entry>
13    <entry>
14      <stringBits>/string<
15      <int>14</int>
16    </entry>
17  </elementAttributes>
18  <visualElement>
19    <elementName>In</elementName>
20    <elementAttributes>
21      <entry>
22        <stringLabel>/string<
23        <stringProgIn>/string<
24        <int>14</int>
25      </entry>
26      <entry>
27        <stringBits>/string<
28        <int>14</int>
29      </entry>
30    </elementAttributes>
31    <visualElement>
32      <elementName>Clock</elementName>
33      <elementAttributes>
34        <entry>
35          <stringLabel>/string<
36          <stringCk>/string<
37        </entry>
38        <entry>
39          <stringLabel>/string<
40          <stringCk>/string<
41        </entry>
42      </elementAttributes>
43      <pos x="340" y="320"/>
44    </visualElement>
45  </visualElement>

```

Latest Submission Status Refresh

# 2248044 (48 tries) compiler msg  
less than a minute ago (08:47:28)  
100.0 [PPPPPPP...]

Helper Refresh [What's this?]

-- There is no comment for this submission --

© 2025 cafe-grader | Revision: 4.1.3cu

## ผ่านหมวดอันดับ 6

ตั้งใจดีอุ

- 6833083521 นางสาว ณัฐณิชา อินทอง
- 6833269021 นาย สิริวิชญ์ ตั้งตรงทรัพย์
- 6833272821 นาย สุรศิทธิ์ ประเสริฐสังข์
- 6833298121 นางสาว ไอริณ ถิระพาณิช

The screenshot shows a web-based grader interface for a CEDT Final Project. The main area displays the XML code for the project. The right side features a summary box with the following details:

- Latest Submission Status: 100.0 [PPPPPPP...]
- # 2248054 (16 tries) compiler msg
- less than a minute ago (08:50:54)

Below the status box, there's a 'Helper' section with a link to 'What's this?' and a note stating "There is no comment for this submission --".

## ผ่านหมวดอันดับ 7

King Opor I

6833228221 Maethasit Setteemongkolchai

6833183621 Palathip Singchoo

6833278621 Hiran Matsayavanit

6833279221 Achitpol Suwankuha

The screenshot shows a web-based application for managing submissions. At the top, there are several tabs including "CEDT Final Project 2568.D", "TEMPLATE Project 2568.d", "worksheet: Link Certificate", "CEDT Grader", "2110252 (2025/1) myCourse", and "CEDT Final Project 2568.D". The main content area has a blue header bar with "CEDT Submissions" and a "File" button. Below this is a large text area containing XML code for a circuit. To the right of the code is a sidebar with "Latest Submission Status" showing a recent submission with ID # 2248062 (68 tries), compiler msg, less than a minute ago (08:52:47), and a score of 100.0. The sidebar also includes a "Helper" section with a refresh button and a note that there is no comment for this submission.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <circuit>
3   <version>2</version>
4   <attributes>
5     <entry>
6       <string>romContent</string>
7       <romList>
8         <rom>
9           <romList>
10          </romList>
11        </attributes>
12      <visualElements>
13        <visualElement>
14          <elementName>In</elementName>
15          <elementAttributes>
16            <entry>
17              <string>Label</string>
18              <string>start</string>
19            </entry>
20            <elementAttributes>
21              <pos x="20" y="560"/>
22            </elementAttributes>
23          <visualElement>
24            <elementName>In</elementName>
25            <elementAttributes>
26              <entry>
27                <string>Label</string>
28                <string>result</string>
29              </entry>
30              <elementAttributes>
31              <pos x="60" y="400"/>
32            </visualElement>
33            <visualElement>
34              <elementName>RAMDualAccess</elementName>
35              <elementAttributes>
36                <entry>
37                  <string>AddrBits</string>
38                  <int>8</int>
39                </entry>
40              <entry>

```

© 2025 cafe-grader | Revision: 4.1.3cu

## ผ่านหมดอันดับ 8

Term ni pi kor S

6833028021 Jiranin Piriyawat (Main Circuit Developer)

6833065221 Nattapong Lumphutta (Main Circuit Developer)

6833248821 Verawood langvong (Main Documentary)

6833125221 Terawee Chaijaroen (Main Documentary)

The screenshot shows a web-based grader interface for a digital logic project. The top navigation bar includes tabs for 'CEDT Grader' (active), 'My Courses' (2025/1), 'CEDT Final Project 2568 Digital', and 'TEMPLATE Project 2568.dig'. The URL is 'cedt-grader.nattree.net/submissions/2248073/edit'. The main content area has a blue header bar with 'CEDT Submissions' and a star icon. The title of the submission is '[DigLo68\_Project\_Full] DigLo68\_Project\_Full'. Below the title, it says 'Limit: 30.0, 51MB'. The code editor contains the following XML code:

```
<dig>
  <version>1.0</version>
  <circuit>
    <version>2</version>
    <ports>
      <visualElements>
        <visualElement>
          <elementName>Out</elementName>
          <elementAttributes>
            <entry>
              <stringLabel></string>
              <stringValue></string>
            </entry>
          </elementAttributes>
        </visualElement>
      </visualElements>
    </ports>
    <elements>
      <visualElements>
        <visualElement>
          <elementName>Out</elementName>
          <elementAttributes>
            <entry>
              <stringLabel></string>
              <stringValue></string>
            </entry>
          </elementAttributes>
        </visualElement>
        <visualElement>
          <elementName>D_FF</elementName>
          <elementAttributes>
            <entry>
              <pos x="2540" y="680"/>
            </entry>
          </elementAttributes>
        </visualElement>
        <visualElement>
          <elementName>D_FF</elementName>
          <elementAttributes>
            <entry>
              <pos x="1560" y="480"/>
            </entry>
          </elementAttributes>
        </visualElement>
        <visualElement>
          <elementName>D_FF</elementName>
          <elementAttributes>
            <entry>
              <pos x="1560" y="560"/>
            </entry>
          </elementAttributes>
        </visualElement>
        <visualElement>
          <elementName>D_FF</elementName>
          <elementAttributes>
            <entry>
              <pos x="1560" y="640"/>
            </entry>
          </elementAttributes>
        </visualElement>
      </visualElements>
    </elements>
  </circuit>
</dig>
```

The right sidebar displays the 'Latest Submission Status' for submission # 2248073 (14 tries) with a 'compiler.msg' file, less than a minute ago at 08:56:51, and a score of 100.0 [PPPPPPP...]. It also shows a 'Helper' section with a 'What's this?' link and a note stating there is no comment for this submission.

Below the code editor, a message box says: 'Write your code in the box above and then click the \*\*\*Submit\*\*\* button to submit the code. This problem requires that the submission must be in **Digital** only.'

At the bottom, there are buttons for 'Language' (set to 'Digital'), 'Choose File', 'No file chosen', and 'Submit'.

ผ่านหมวดอันดับ 9

## Group : Withdraw

6833192221 Patcharapong Prapanwattana  
6833070321 Napatr Boonyarangkul  
6833066921 Nathaphong Wongwiangchan

CEDT Submissions ▾

[DigLo68\_Project\_Full] DigLo68\_Project\_Full

Limit: 30.0s, 511MB

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <circuit>
3   <version>2</version>
4   <attributes/>
5   <visualElements>
6     <visualElement>
7       <elementName>Tunnel</elementName>
8       <elementAttributes>
9         <entry>
10          <string>rotation</string>
11          <rotation rotation="2"/>
12        </entry>
13        <entry>
14          <string>NetName</string>
15          <string>command</string>
16        </entry>
17      </elementAttributes>
18      <pos x="2720" y="700"/>
19    </visualElement>
20    <visualElement>
21      <elementName>Splitter</elementName>
22      <elementAttributes>
23        <entry>
24          <string>splitterSpreading</string>
25          <int>4</int>
26        </entry>
27        <entry>
28          <string>Input Splitting</string>
29          <string>14</string>
30        </entry>
31        <entry>
32          <string>Output Splitting</string>
33          <string>8,6</string>
34        </entry>
35      </elementAttributes>
36      <pos x="2740" y="700"/>
37    </visualElement>
38    <visualElement>
```

ผ่านหมวดอันดับ 10