01076114
องค์ประกอบและสถาปัตยกรรมคอมพิวเตอร์
Computer Organization and Architecture
01076115
ปฏิบัติการองค์ประกอบคอมพิวเตอร์
COMPUTER ORGANIZATION IN PRACTICE

Introduction, Performance

เนื้อหาของวันนี้

- ทำไมต้องเรียนวิชานี้ (องค์ประกอบคอมพิวเตอร์)
- ข้อกำหนดและข้อตกลง
- ทิศทางใหม่ๆ ของเทคโนโลยีด้านฮาร์ดแวร์

ทำไมต้องเรียน Computer Organization

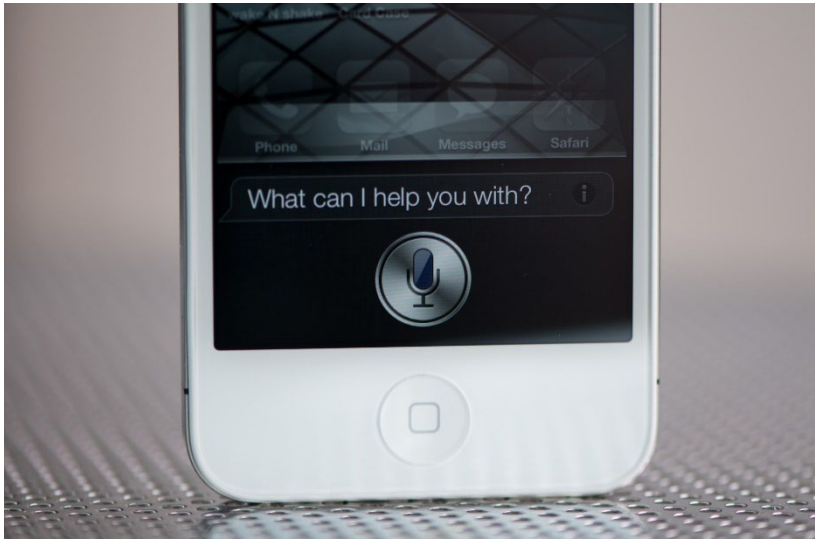


Image credits: uber, extremetech, anandtech

ทำไมต้องเรียน Computer Organization

- คงเสียฟอร์ม ถ้าวิศวะคอมพิวเตอร์ ลาดกระบัง ไม่สามารถอธิบายคำต่อไปนี้ได้ DRAM, pipelining, cache hierarchies, virtual memory, อื่นๆ
- คงเสียฟอร์ม ถ้าวิศวะคอมพิวเตอร์ ลาดกระบัง ไม่สามารถบอกเพื่อนได้ว่าจะเลือก โพรเซสเซอร์ ตัวไหนดี (อย่างมีหลักวิชาการ) สำหรับคอมพิวเตอร์ที่จะซื้อ
- การรู้ฮาร์ดแวร์จะช่วยให้เขียนโปรแกรมได้ดีขึ้นหรือปลอดภัยขึ้นมั้ย?
- วิชานี้เป็นพื้นฐานของวิชา OS

ผมจะเป็น Dev ผมไม่ต้องรู้ hardware หรอก

- ในบางครั้ง Dev จำเป็นต้องรู้ว่าจะเขียนโปรแกรมให้มีประสิทธิภาพที่ดีกว่าได้อย่างไร เช่น multi-core processor
- การรู้ hardware จะทำให้สามารถเขียนโปรแกรมได้ปลอดภัยมากขึ้น
- ในการพัฒนาในงานบางด้าน เช่น IoT จำเป็นต้องรู้เกี่ยวกับ hardware
- การเข้าใจ hardware จะทำให้รู้ว่าข้อมูลอยู่ที่ไหน เพื่อจัดการให้ข้อมูลที่เกี่ยวข้องกัน อยู่ใกล้กัน
- ความเข้าใจเรื่อง thread ทำให้สามารถเขียนโปรแกรม multi-thread ได้ดีขึ้น (ทำไมต้องเขียน multi-thread ด้วย?)

ตัวอย่างการปรับปรุงโปรแกรมเพื่อประสิทธิภาพที่ดีขึ้น

200x speedup for matrix vector multiplication

- Data level parallelism: 3.8x
- Loop unrolling and out-of-order execution: 2.3x
- Cache blocking: 2.5x
- Thread level parallelism: 14x

คำอธิบายรายวิชา

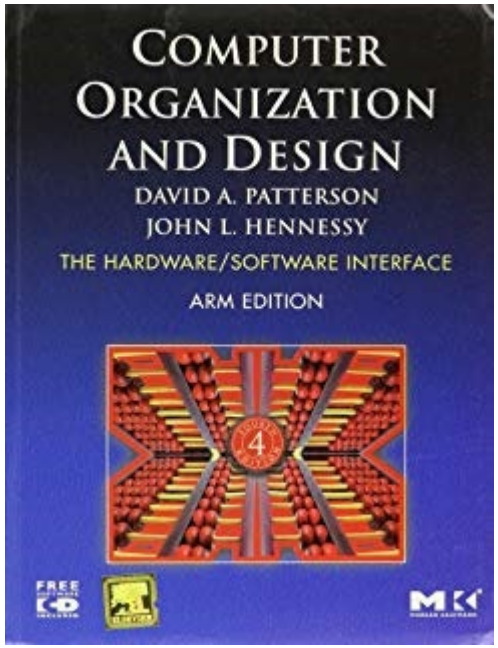
- ภาพรวมขององค์ประกอบและสถาปัตยกรรมคอมพิวเตอร์ การแทนข้อมูลในคอมพิวเตอร์ การจองและเข้าถึงหน่วยความจำ หน่วยประมวลผลกลาง การเขียนโปรแกรมภาษาแอสเซมบลีและสถาปัตยกรรมชุดคำสั่ง การทำงานของซอฟต์แวร์ระดับสูงในมุมมองของชุดคำสั่งระดับล่าง ระดับชั้นของหน่วยความจำ เทคนิคการส่งข้อมูลและอินพุตเอาท์พุต การคำนวณของคอมพิวเตอร์ การวัดประสิทธิภาพของระบบ
- Overview of Computer Architecture and Organization; Data Representation, Memory Allocation and Access; Central Processing Unit; Assembly Programming and Instruction Set Architecture; High-level Software to Low-level Instructions; Memory Hierachy; Data Transfer and Input/Output (I/O) Techniques; Computer Arithmetic; Measuring system performance; Towards higher speed

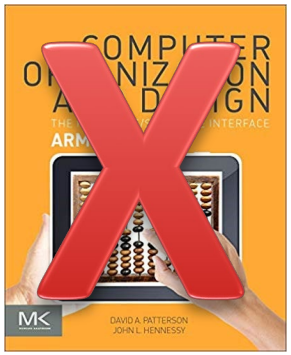
เนื้อหาที่เรียน

- เพื่อให้เข้าใจการทำงานของโปรแกรม ว่ามีอะไรเกิดขึ้นในขณะที่คอมพิวเตอร์ทำงานตามโปรแกรม
- เพื่อให้เข้าใจโครงสร้างการทำงานของระบบคอมพิวเตอร์
- สามารถเขียนโปรแกรมภาษาแอสเซมบลีได้
- เนื้อหา
  - Moore’s Law, power wall
  - Use of abstractions
  - Assembly language
  - Computer arithmetic
  - Pipelining
  - Using predictions
  - Memory hierarchies

ตำรา

- Computer Organization and Design – HW/SW Interface, Patterson and Hennessy, 4th edition, ARM Edition





- มีขายที่ร้านหนังสือหลังธนาคาร

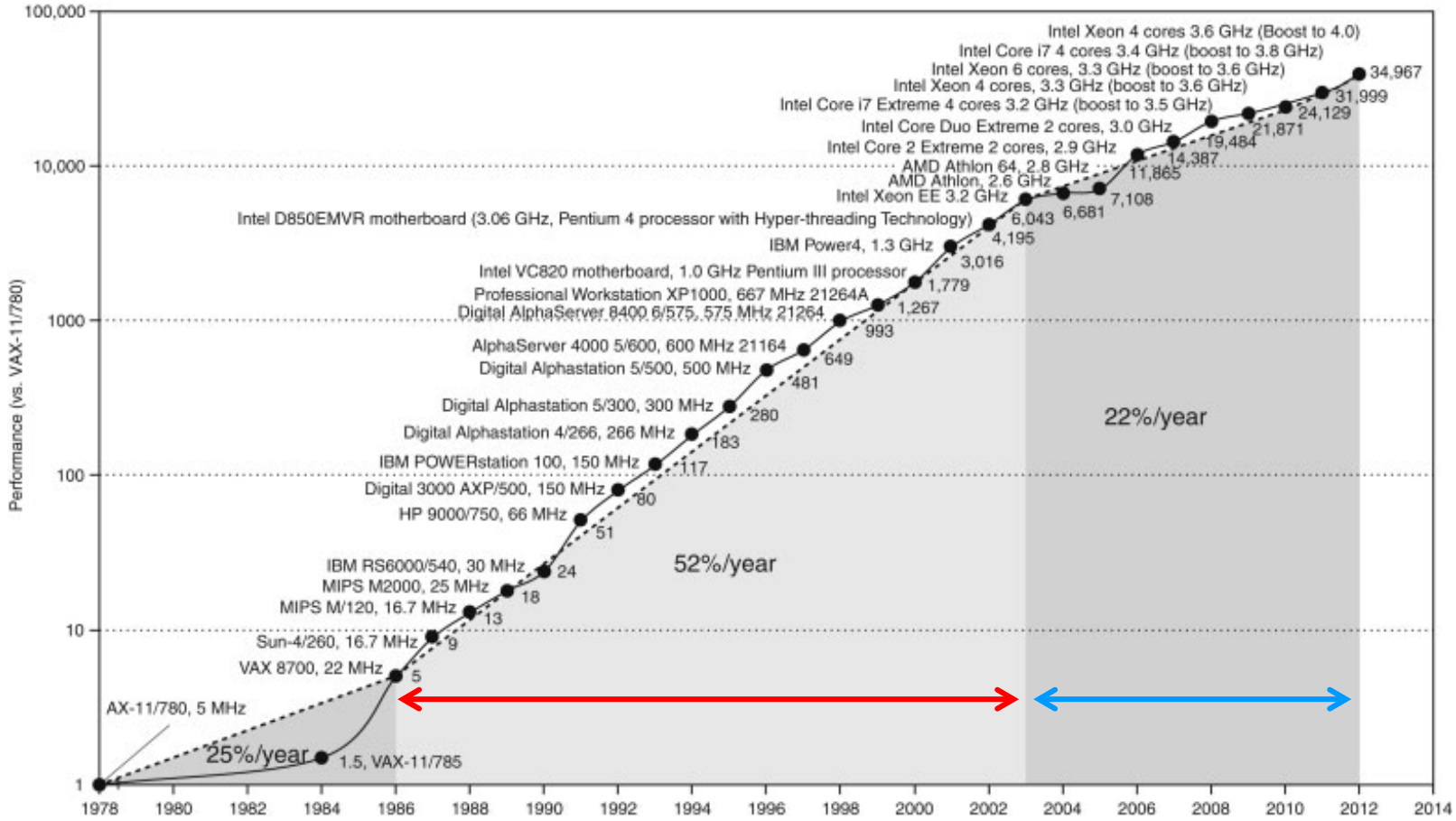
คะแนน Lecture

หัวข้อ	คะแนน
การบ้าน	10
ชิ้นงาน (Assignment)	20
สอบกลางภาค	35
สอบปลายภาค	35

คะแนน Lab

หัวข้อ	คะแนน
ส่ง Lab	45
ชิ้นงาน (Assignment)	30
สอบ Lab	25

Microprocessor Performance



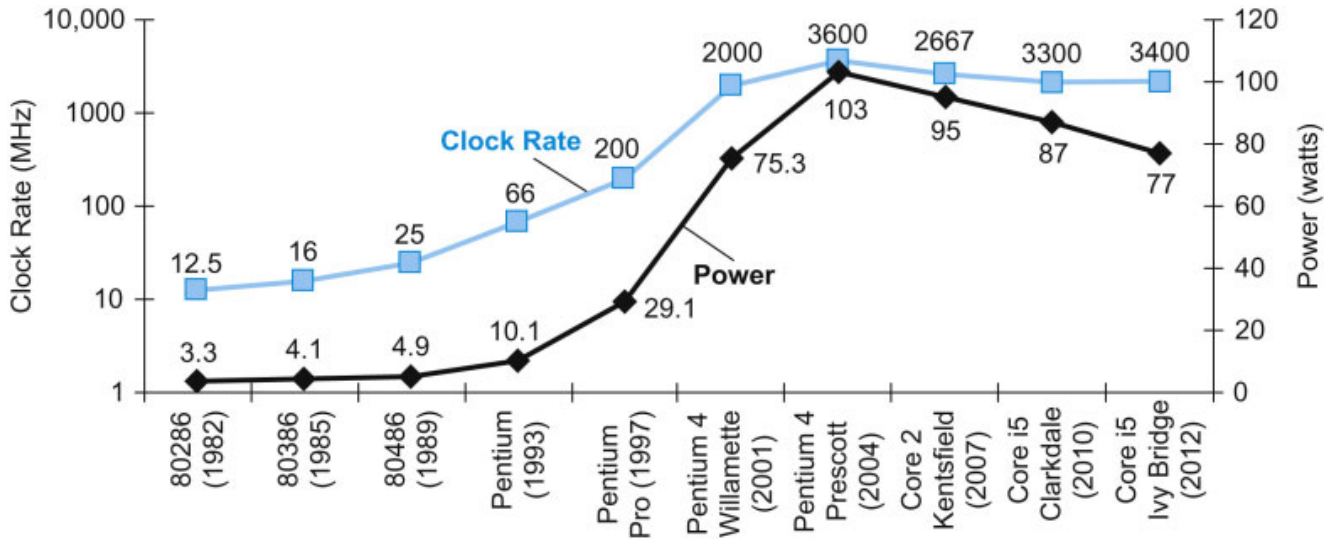
50% improvement every year!!

What contributes to this improvement?

Power Consumption Trends

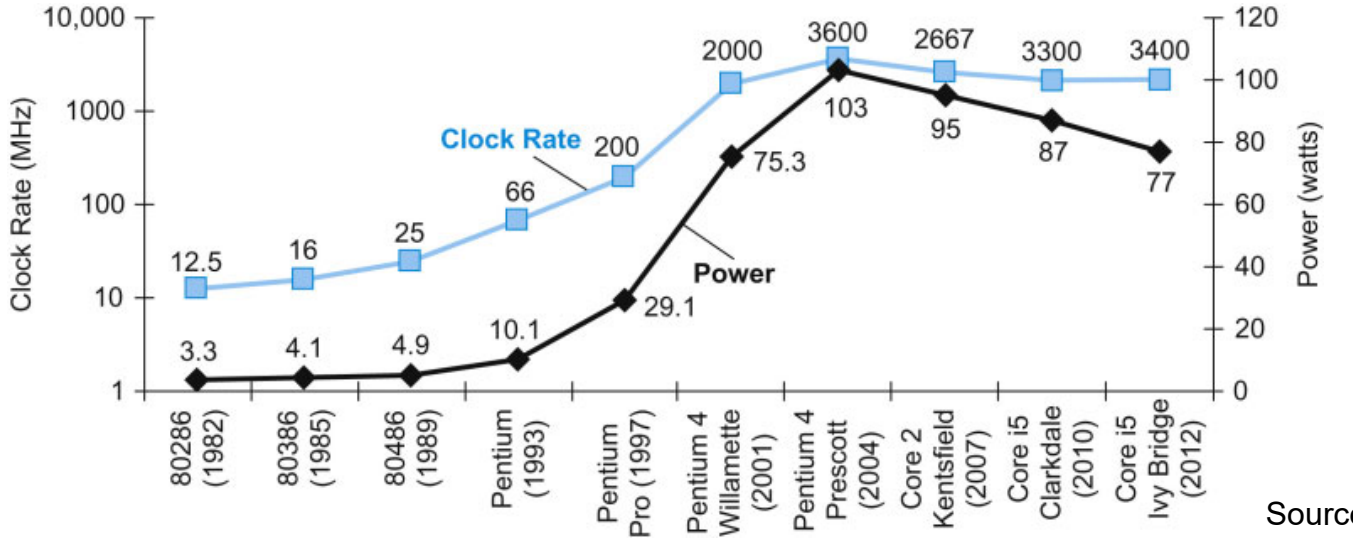
- Dyn power ≈ activity x capacitance x voltage^2 x frequency

- ถ้าแรงดันและความถี่มีค่าเท่าเดิม แต่ปรับปรุงให้ Transistor มีขนาดเล็กลง (จำนวนเพิ่ม) จะทำให้ทำงานได้มากขึ้น (activity เพิ่ม) และประจุลดลง (ขนาดเล็กลง)
- ถ้าลดแรงดัน จะทำให้ใช้พลังงานน้อยลง ทำให้สามารถเพิ่มความถี่ในการทำงานได้



Power Consumption Trends

- แม้จะลดแรงดันแล้ว (1.2v) แต่ด้วยความจุต่อพื้นที่มาก ทำให้เกิดความร้อนมาก
- ทำให้ต้องเพิ่มการระบายความร้อนด้วย heat sink และพัดลม (และวิธีอื่นๆ)
- ปัจจุบันโพรเซสเซอร์จะทำงานได้ เมื่อค่าพลังงานไม่เกิน 100 วัตต์
- การลดแรงดัน การเพิ่มความถี่ หรือ การลดขนาด ทำได้ยากมากขึ้น



Source: H&P Textbook

Important Trends

- ผลคือ เริ่มหมดหนทางในการปรับปรุงประสิทธิภาพของโพรเซสเซอร์ (สำหรับ single thread)
- ปัญหา Power wall ที่กล่าวมา ทำให้ยากต่อการเพิ่มประสิทธิภาพของโพรเซสเซอร์อีก
- และยากต่อการจะเพิ่ม clock speed (จะเห็นว่า clock speed ติดอยู่ประมาณ 3 GHz มานานแล้ว)

Important Trends

- แนวทางการปรับปรุงประสิทธิภาพในอดีต
  1. Better processes (faster devices) ~20%
  2. Better circuits/pipelines ~15%
  3. Better organization/architecture ~15%
- ในอนาคต ข้อ 1. จะทำไม่ได้อีก และข้อ 2. ยังช่วยได้นิดหน่อย

	Pentium	P-Pro	P-II	P-III	P-4	Itanium	Montecito
Year	1993	95	97	99	2000	2002	2005
Transistors	3.1M	5.5M	7.5M	9.5M	42M	300M	1720M
Clock Speed	60M	200M	300M	500M	1500M	800M	1800M

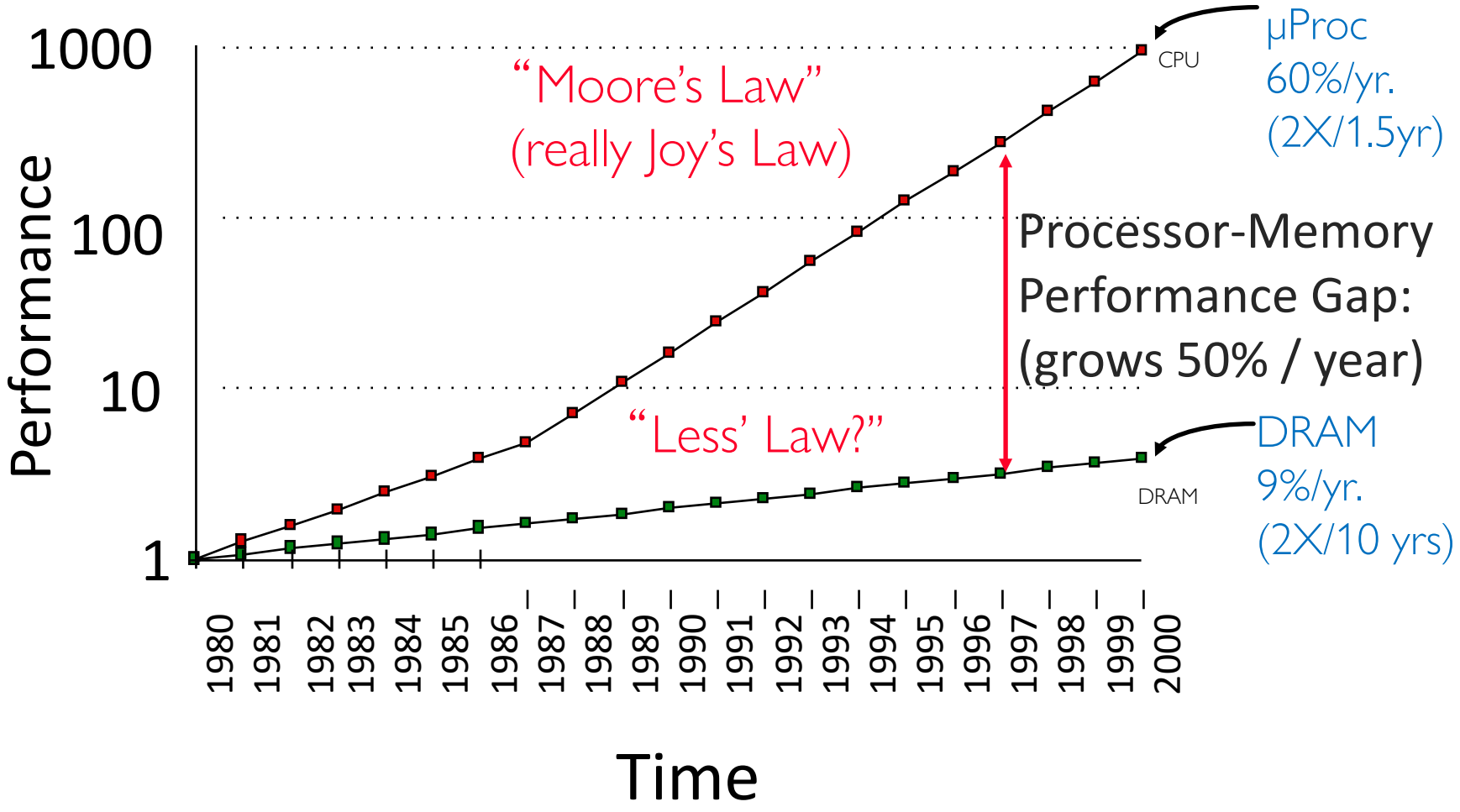
Moore's Law in action

At this point, adding transistors to a core yields little benefit

Memory and I/O Technology Trends

- ความจุของหน่วยความจำ (DRAM) เพิ่มขึ้นประมาณ 40-60% ต่อปี แต่ latency กลับลดลงเพียง 9% ในระยะเวลา 10 ปี (memory wall!) (ต้องใช้ multi channel เข้ามาช่วย)
- ความจุของดิสก์ไดรฟ์ เพิ่มขึ้น 100% ทุกปี แต่การลด latency ทำได้ในอัตราพอๆ กับ DRAM
- ระบบเครือข่าย ปัจจุบันแบนด์วิธ 1 Gbps กลายเป็นมาตรฐานไปแล้ว และกำลังจะกลายเป็น 10 Gbps ในอนาคต

Processor-DRAM Memory Gap (latency)



What Does This Mean to a Programmer?

- การพัฒนาประสิทธิภาพทำได้เพียง 20 % ในแต่ละปี และจะน้อยกว่านี้หากโปรแกรมไม่เขียนเป็น multi-threads
  - โปรแกรมจะใช้ thread มากขึ้น
  - เมื่อใช้ thread มากขึ้น ก็ต้องการ synchronization และ communication ระหว่าง thread ที่ดีขึ้น
  - ตำแหน่งของข้อมูลที่อยู่ในหน่วยความจำ จะเป็นปัจจัยสำคัญต่อประสิทธิภาพมากขึ้น
  - ต้องมีการเรียกใช้ Accelerators ให้มากเท่าที่จะเป็นไปได้

The HW/SW Interface

Application software

Systems software
(OS, compiler)

Hardware

```
a[i] = b[i] + c;
```

Compiler

```
lw    $15, 0($2)
add   $16, $15, $14
add   $17, $15, $13
lw    $18, 0($12)
lw    $19, 0($17)
add   $20, $18, $19
sw    $20, 0($16)
```

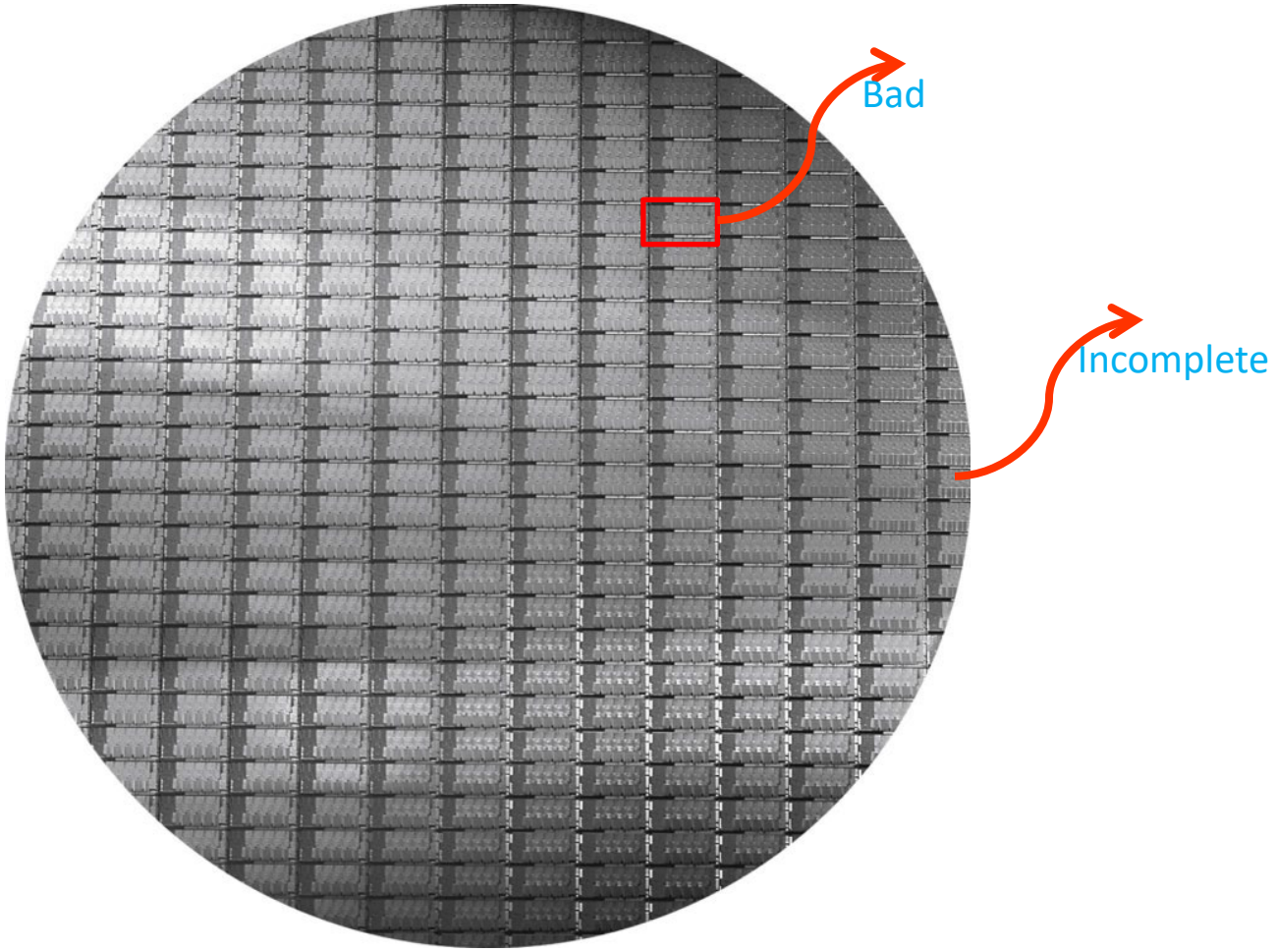
Assembler

```
00000010110000
11010000010010
…
```

Computer Components

- Input/output devices
- Secondary storage: non-volatile, slower, cheaper
- Primary storage: volatile, faster, costlier
- CPU/processor (datapath and control)

Wafers and Dies



Source: H&P Textbook

Manufacturing Process

- ในการผลิต Silicon Wafers มีหลายขั้นตอนมาก ทั้งประกอบด้วยส่วนของฉนวน ส่วนของตัวนำ และสารกึ่งตัวนำ (transistor)
- Wafers จะถูกตัดออกเป็นชิ้นสี่เหลี่ยม เรียกว่า die ซึ่งขนาดของ die จะเป็นตัวกำหนด yield และราคา
- Yield คือ ผลที่จะได้จาก Wafers 1 แผ่น โดยหักส่วนขอบที่ใช้ไม่ได้ และ die ที่เสียออกไปแล้ว
- ดังนั้นยิ่ง die มีขนาดเล็กก็จะทำให้ Yield ยิ่งมาก

Processor Technology Trends

- ขนาดของ transistor มีการลดลงอย่างต่อเนื่อง : 250nm (1997) → 130nm (2002) → 70nm (2008) → 35nm (2014) → 22nm (2016) → 14nm (2018)
- ความหนาแน่นของ transistor เพิ่มขึ้นประมาณ 35 เปอร์เซ็นต์ต่อปี และ die size เพิ่มขึ้น 10-20 เปอร์เซ็นต์ต่อปี
- ความเร็วในการทำงานของ transistor มีความสัมพันธ์กับขนาดมาก ยิ่งผลิตได้ที่ขนาดเล็ก จะยิ่งทำงานได้เร็วขึ้น
- แต่ Wire delay ไม่ได้ลดลงเร็วเท่ากับ transistor เนื่องจากความต้านทานของสายไฟ

Performance Metrics

- การวัดประสิทธิภาพ
  - response time เวลาที่ใช้ตั้งแต่เริ่มต้นโปรแกรมจนจบโปรแกรม
  - Throughput จำนวนงานที่ทำได้ในช่วงเวลาที่กำหนด
- ในการวัดทั้งสองแบบโดยทั่วไปจะมีความสัมพันธ์กัน
  - ถ้าโพรเซสเซอร์มีความเร็วมากขึ้น ทั้ง response time และ throughput จะดีขึ้นทั้งคู่
  - แต่ถ้าเพิ่มจำนวนโพรเซสเซอร์ จะมีแต่ throughput ที่จะดีขึ้น
  - บางสถานการณ์ อาจทำให้ throughput ดีขึ้นแต่ response time แย่ลง (เช่น มี 2 Core รันโปรแกรมเดียวกัน แต่แชร์หน่วยความจำเดียวกัน)

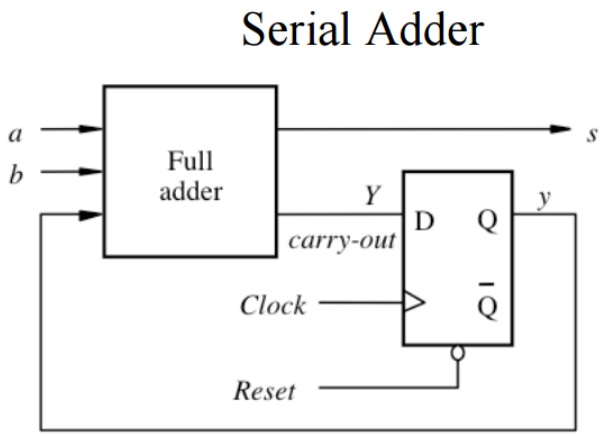
Execution Time

- กำหนดให้ระบบ X รับงาน W มาทำ
  - $\text{Performance}_X = 1 / \text{Execution time}_X$
- Execution time = response time = wall clock time
  - Execution time รวมทั้งเวลาที่โปรแกรมทำงานและเวลาที่ใช้โดยระบบปฏิบัติการ
- สมมติว่าระบบ X รันโปรแกรมเสร็จใน 10 วินาที ระบบ Y รันโปรแกรมเสร็จใน 15 วินาที
  - ระบบ X เร็วกว่าระบบ Y 1.5 เท่า
  - perf X / perf Y = exectime Y / exectime X
  - ประสิทธิภาพของ X เหนือกว่า Y

    1.5 -1 = 0.5 = 50% = (perf X – perf Y) / perf Y = speedup - 1
  - ถ้าจะให้ X รันช้าลงให้เท่ากับ Y จะต้องลดลง (15-10) / 15 = 33%
  - ถ้าจะใช้ Y รันเร็วขึ้นให้เท่ากับ X จะต้องเพิ่มขึ้น (15-10) / 10 = 50%

Clocks and Cycles

- ในระบบคอมพิวเตอร์จะมีวงจรย่อยๆ ทำงานร่วมกันอยู่ภายใน เพื่อทำงานร่วมกันเป็น 1 คำสั่ง เช่น คำสั่งบวกเลข ก็ต้องมีวงจรบวก และวงจรนำข้อมูลเข้าออก



- ในการทำงาน 1 คำสั่ง มักออกแบบให้ทำงานตามจังหวะของสัญญาณนาฬิกา เพื่อให้ส่วนต่างๆ ประสานกันได้ดี
- สมมติว่าในคำสั่งบวก วงจรบวกใช้เวลามากที่สุด คือ 800 ps ดังนั้น clock ก็จะต้องมีค่าไม่เกิน 1/800 ps = 1.25 GHz เราเรียกค่านี้ว่า CPI (Cycle per Instruction) (กรณีนี้ คือ 1 CPI)

Performance Equation - I

CPU execution time = CPU clock cycles x Clock cycle time
Clock cycle time = 1 / Clock speed

- โพรเซสเซอร์ตัวหนึ่งมีความถี่ 3 GHz (ใน 1 วินาทีจะมี 3 พันล้าน clock tick) ในแต่ละ clock tick จะมีคำสั่งที่ทำงานเสร็จกี่คำสั่ง?
- ถ้าโปรแกรมหนึ่ง ต้องทำงานเป็นระยะเวลา 10 วินาทีจึงจะเสร็จ บนโพรเซสเซอร์ความเร็ว 3 GHz โปรแกรมนี้ต้องใช้กี่ Clock cycle จึงทำงานเสร็จ?
- ถ้าโปรแกรมหนึ่งต้องใช้ 2 พันล้าน Clock cycle ในโพรเซสเซอร์ความเร็ว 1.5 GHz จงหา execution time (ตอบเป็นวินาที)

Performance Equation - II

CPU clock cycles = no. of instructions x avg. clock cycles per instruction (CPI)

- เมื่อแทนสมการข้างต้นลงในสมการในหน้าที่แล้ว จะได้ว่า

Execution time = clock cycle time x no. of instructions x avg. CPI

- โพรเซสเซอร์ความเร็ว 2 GHz ตัวหนึ่ง ทำงาน 1 คำสั่งใช้ 3 clock cycles ในเวลา 10 วินาที โพรเซสเซอร์ตัวนี้จะทำงานได้กี่คำสั่ง

Factors Influencing Performance

Execution time = clock cycle time x no. of instructions x avg. CPI

- Clock cycle time : ขึ้นกับเทคโนโลยีการผลิตของผู้ผลิต ที่จะทำให้ transistor เร็วได้แค่ไหน และจะทำให้ pipeline มีประสิทธิภาพแค่ไหน (later)
- No. of instructions : ขึ้นกับความสามารถของ compiler และ Instruction set Architecture (คำสั่งเก่งหรือไม่เก่ง)
- CPI : ขึ้นกับการทำงานของแต่ละคำสั่ง และความสามารถในการออกแบบสถาปัตยกรรมของโพรเซสเซอร์

Example

Execution time = clock cycle time x no. of instructions x avg. CPI

- มีระบบคอมพิวเตอร์อยู่ 2 ระบบ รันโปรแกรมเดียวกัน ให้บอกว่าระบบคอมพิวเตอร์ใด เร็วกว่า?
  - ระบบแรก ใช้โพรเซสเซอร์ MIPS เมื่อโปรแกรมนี้ผ่าน compiler จะได้คำสั่งภาษาเครื่อง (MIPS Instruction) ออกมา 4 ล้านคำสั่ง โดยโพรเซสเซอร์ MIPS จะทำงานแต่ละคำสั่งเสร็จโดยเฉลี่ย 1.5 clock cycles (CPI) โดยระบบนี้ทำงานที่ความถี่ 1 GHz
  - ระบบที่สอง ใช้ X86 เมื่อโปรแกรมเดียวกันผ่าน compiler จะได้คำสั่งภาษาเครื่อง (X86 Instruction) ออกมา 2 ล้านคำสั่ง โดยโพรเซสเซอร์ X86 จะทำงานแต่ละคำสั่งเสร็จโดยเฉลี่ย 6 clock cycles (CPI) โดยทำงานที่ความถี่ 1.5 GHz

Power and Energy

- Total power = dynamic power + leakage power
- Dynamic power = activity x capacitance x voltage^2 x frequency
- Leakage power = voltage
- Energy = power x time

  (joules) (watts) (sec)

Example

- โพรเซสเซอร์ตัวหนึ่งทำงานที่ความถี่ 1 GHz ใช้เวลา 100 วินาที ในการรันโปรแกรมโดยใช้ Dynamic Power 70W และ Leakage Power 30W โพรเซสเซอร์ตัวนี้มีโหมด Turbo Boost ด้วย โดยเมื่อทำงานในโหมดนี้ จะเพิ่มความถี่เป็น 1.2 GHz เมื่อโพรเซสเซอร์ตัวนี้รันโปรแกรมเดิมในโหมด Turbo Boost จะใช้พลังงานเพิ่มขึ้นหรือลดลง?

Example

- โพรเซสเซอร์ตัวหนึ่งทำงานที่ความถี่ 1 GHz ใช้เวลา 100 วินาที ในการรันโปรแกรมโดยใช้ Dynamic Power 70W และ Leakage Power 30W โพรเซสเซอร์ตัวนี้มีโหมด Turbo Boost ด้วย โดยเมื่อทำงานในโหมดนี้ จะเพิ่มความถี่เป็น 1.2 GHz เมื่อโพรเซสเซอร์ตัวนี้รันโปรแกรมเดิมในโหมด Turbo Boost จะใช้พลังงานเพิ่มขึ้นหรือลดลง?
- Normal mode energy = 100W x 100s = 10,000 J
- Turbo boost energy = (70 x 1.2 + 30) x 100/1.2 = 9,500 J

Power and Energy

$$\text{Dyn Power} = \text{Capacitive Load} \times \text{Voltage}^2 \times \text{Frequency}$$

- สมมติว่าเราพัฒนาโพรเซสเซอร์ตัวใหม่ โดยมี Capacitive Load ลงลงเหลือ 85% และลด voltage ลง 15% ผลกระทบของ Power จะเป็นเท่าไร?

SPEC

- ในการเปรียบเทียบระบบคอมพิวเตอร์ที่ใช้ Instruction Set เดียวกัน เราสามารถใช้ CPI ในการเปรียบเทียบได้ เช่น ระหว่าง Intel กับ AMD หรือตระกูล ARM (เช่น Snapdragon กับ Mediatek) แต่สำหรับระบบคอมพิวเตอร์ที่มีชุดคำสั่งต่างกัน จะเปรียบเทียบกันโดยใช้ CPI ไม่ได้
- ผู้ผลิตหลายรายจึงรวมกันสร้างมาตรฐานสำหรับเปรียบเทียบ Performance ขึ้นมา โดยตั้งชื่อว่า SPEC(Standard Performance Evaluation Corporation) ซึ่งเป็นชุดโปรแกรมที่มีการประมวลผลหลายๆ อย่าง เช่น จำนวนเต็ม ทศนิยม และอื่นๆ
- ผู้ผลิตแต่ละรายจะนำโปรแกรมไปรัน และประกาศเป็น SPEC Rating ทำให้สามารถเทียบกันได้
- SPEC2017 (ล่าสุด) ประกอบด้วยโปรแกรม 43 ชุด รายละเอียดดูที่ https://www.spec.org/cpu2017/Docs/overview.html#suites

Exercise #1

- ให้ทำ Exercise 1.3 & 1.4 กำหนดส่ง 1 สัปดาห์
- เขียนด้วยลายมือเท่านั้น
  - ทำลงบนกระดาษ แล้ว Scan หรือ ถ่ายรูป
  - ทำลงใน Tablet แล้ว Export
- ส่ง PDF 1 ไฟล์ ผ่าน Forms

For your attention