Memory Function

Memory --> Store

Program --> เก็บใน Code segment ของ main memory
Data --> เก็บใน Data Segment ของ Main Memory
State --> เก็บใน Stack Segment ของ Main Memory
Free Memory (ที่ว่าง) --> เก็บไว้ใช้งานส่วนต่างๆ

Stack Operation

Keep Status or Address of Calling Program
Last-In First-Out
First-In Last-Out

ก่อนจะทำงาน subprogram จะมีการบันทึก ตำแหน่งคำสั่งตัวถัดไปของ PC ในตัว CPU เก็บไว้ใน Stack

Stack Pointer เป็นตัวชี้ของ Stack บนสุด
ทุกครั้งที่ pop หรือ push ค่า Stack Pointer จะเปลี่ยนเสมอ

Parameter Passing by Stack

- การแลกเปลี่ยน parameter ปัจจุบันจะใช้ stack ในการถ่ายโอนค่า parameter
- เอาค่า parameter ตัวสุดท้าย push ลง stack แล้วค่อย call function
- เมื่อ Execute ค่า Parameter จะถูกเรียกผ่านตำแหน่งของ Stack แล้วเอาไปบวกกับตำแหน่งที่ถูกเก็บไว้
- ยัดตัวแปร, return address และ method ที่กำลังทำตามลำดับ

Cache Memory

Memory ช้ากว่า CPU
CPU wait Memory --> Slow

Cache

- มี 3 level
- 1 เร็วสุด ใกล้ CPU สุด, 3 ช้าสุด ไกล CPU สุด
- Cache --> เร็ว --> เก็บจำนวนได้น้อย
- Locate: ติดตั้งอยู่ระหว่าง CPU และ Memory
- Speed: ความเร็วระหว่าง CPU และ Memory

Cache / Main Memory Structure

Cache จะจุน้อยกว่า Memory
Memory จะเก็บข้อมูลเป็น block ๆ
Cache จะเก็บเป็น line (block but cache) และจะมี tag อยู่ใน line (บอกว่าอยู่ส่วนไหนใน main memory)

Cache Operation

เมื่อ CPU ต้องการทำงานกับ Main Memory จะต้องเรียกผ่าน Address

CPU ก็จะส่งข้อมูลไปที่ Cache Controller เพื่อหาว่ามี Address ที่ CPU ต้องการไหม

ถ้ามีจะเป็น Cache Hit เอาไปใช้ได้เลย เร็ว
ถ้าไม่มีจะเป็น Cache Miss ต้องไปเรียกใน Main Memory

ถ้า Cache Miss, controller จะเก็บ address ที่ร้องขอจาก Main memory มาไว้ใน cache

ต้องออกแบบให้มี Cache Hit ได้มากที่สุด เพื่อประสิทธิภาพสูงสุด

Cache Write Operation

ข้อมูลใน Cache และ Main Memory ต้องเหมือนกัน เวลาจะ Write (ข้อมูลทุกที่ต้องเหมือนกัน)

1. Write Through: เขียน Cache กับ Main Memory พร้อมกัน --> ช้า แต่โอกาสผิดพลาดน้อย
2. Write Back: เขียนแค่ Cache แล้วพอ Cache ไม่ได้ใช้แล้ว Cache จึงไปอัพเดท Main Memory เอง --> เร็วกว่า แต่เกิดข้อผิดพลาดข้อมูลไม่ตรงกันสูงกว่า

Cache Mapping Function

Mapping เพื่อให้รู้ว่า cache นี้อยู่ไหนใน Main Memory

Direct Mapping:
Map Cache กับ Memory ผ่าน Index เป็นตัวกำหนดข้อมูลใน Cache

แบ่งเป็น 3 ส่วนคือ

- Word คือขนาดของข้อมูลใน Block จำนวนบิตจะบอกขนาด Block (2^n Size)
- Line คือตำแหน่งของข้อมูลแต่ละตัวใน Block
- Tag คือ Memory Address ที่ใช้อ้างอิง

Fully Associative Mapping
- Tag บอกพื้นที่ทั้งผืนของ Memory
- อนุญาตให้ block ใดๆ ของ main memory map ไปยัง cache line ที่ว่างอยู่ได้

Set Associative Mapping

- มีกี่เซ็ต เป็นตัวเลข k --> k-way set associative mapping
- ส่วนมาก k เป็นเลข 2^n

Two-Way set associative mapping

- ซับซ้อนกว่า Direct Mapping และประสิทธิภาพดีกว่า Fully Associative
- แบ่งเป็นช่องละ 2 set
- ข้อมูลจะถุกแบ่งเป็น 2 ชุด เอาแค่ set ที่เลือกไว้ไปหา

Cache Replacement Algorithm

Cache เต็มต้องเอาข้อมูลใหม่มาแทนที่ ทำไงดี

Write Through --> ช่างแม่ง, Write Back --> ต้องแทนที่

Least Recently Used (RLU) Policy

- เอาข้อมูลล่าสุดที่ไม่ค่อยได้ใช้มากสุด ออก

First In First Out

- เข้ามาก่อน ก็เอาออกก่อน

Least Frequently Used

- มีการใช้งานน้อยสุด เอาออก และต้องมีการบันทึกว่า block ไหนใช้ตัวไหนไปบ้าง

Random

- สุ่มมั่ว

Storage Hierarchy

คอมทำงานเป็นลำดับชั้น
ความเร็วสูง --> แพง

ลำดับชั้น

- Registers
- L1 Cache
- L2 Cache
- L3 Cache

- Main memory Disk cache
- Solid State Disk (SSD) Hard Disk
- Optical Disk
- Tape

Storage Types

Internal Storage

- Register
- Cache
- Memory

Storage Types

- ROM
- RAM
- Static RAM - Register, Cache Dynamic
- Dynamic RAM - Main Memory
- Flash Memory - SSD
- Hard Disk
- Optical Disk
- Tape

Physical Storage Types

Semiconductor

- RAM, ROM

Magnetic

- Disk and Tape

Optical

- CD and DVD

Others

- Bubble
- Hologram

Semiconductor

1. Read Only Memory (ROM)
    - Non-volatile
    - ส่วนใหญ่ ใช้เริ่มต้นทำงานคอมเช่น BIOS
    - ROM ส่วนมากเขียนข้อมูลมาจากโรงงานแล้ว
    - PROM เขียนได้รอบเดียว
    - EPROM ล้างได้โดยใช้แสง UV
    - EEPROM ล้างได้โดยแรงดันไฟฟ้าแรงดันสูง (12V, 20V, etc.) (ปัจจุบันก็อาจต่ำลงหน่อย)
    - Flash Memory --> ใช้แรงกันไฟฟ้าปกติในการทำงาน --> ปัจจุบันใช้ส่วนใหญ่
2. Random Access Memory (RAM)
    - มี Address Bus แยกออกมา เข้าถึงตำแหน่งได้ด้วย Address
    - เป็น Volatile
    - Static RAM --> เอา transistor มาประกอบกันในลักษณะ flip-flop เพื่อให้คงค่าสถานะข้อมูลไว้
        - เร็วแต่ความจุน้อย กินพื้นที่ แพง
        - 1-bit ใช้ Transistor ไม่ต่ำกว่า 6 ตัว
        - ใช้เป็น Register หรือ Cache
    - Dynamic RAM --> เก็บข้อมูลโดยใช้ประจุไฟฟ้า
        - ทำงานช้ากว่า Static RAM
        - 1-bit ใช้ Transistor เก็บประจุควบคุมแค่ 1 ตัว
        - ทำหน้าที่ Charge/Discharge ตัวเป็นประจุให้เป็นค่า 0 / 1

DRAM Refreshing

- การอัพเดทข้อมูลในหน่อยความจำแบบ DRAM เพื่อป้องกันการเสียข้อมูลที่เก็บไว้
- อัพเดทด้วยการอ่านทั้งหมดแล้วเขียนใหม่หมด
- ถ้าต้องเข้าถึงข้อมูลตอน refreshing ต้องรอก่อน

Memory Error Correction

- เป็นการเก็บ Error Correction Bit เพื่อเช็คว่าข้อมูลถูกปล่าว
- มีพื้นที่เก็บใน Memory
- เมื่อพบข้อผิดพลากจะส่งข้อมูลไปให้ CPU

Synchronous DRAM

- เป็นชนิดของหน่วยความจำ
- ไม่ได้ถุกเควบคุมโดย CPU แต่ผ่าน cache controller แล้วทำไมไม่เชื่อมกับนาฬิกาเลย
- Synchronous กับการทำงานของจังหวะนาฬิกา (clock)

DDR SDRAM (Double Data Rate)

- 1 clock อ่านได้หลายข้อมูล (2, 3, 4, 5 blocks, ฯลฯ)

External Memory

- Magnetic
- Optical