

Input & Output

Overview

- เราไม่สามารถเก็บข้อมูลเยอะๆ ได้ใน CPU และ Main Memory
- CPU เก็บข้อมูลไว้ประมวลผล, Main Memory เอาไว้เก็บชั่วคราว

I/O Functions:

- Transfer Data between computer and External Device
- Input --> Outside to CPU/Memory
- Output --> CPU/Memory to Outside

Internal Locations:

CPU --> Direct to Register

Memory ---> Temporary, Wait for CPU processing

Transferring between Main Memory

Input --> Memory

Memory --> Output

External Devices ทำงานผ่าน I/O Module ซึ่งภายในประกอบด้วย

- Control Logic --> การควบคุม
- Transducer --> การโอนย้ายข้อมูล, มี buffer พักข้อมูล

External Device:

Human readable --> Screen, printer, keyboard

Machine readable --> Monitoring and Control

Communication --> Modem, NIC

I/O Module

- Slower than CPU and Memory
- รวม I/O หลายๆ ตัว หลาย format หลายความเร็ว
- เป็น Interface ให้ทั้ง CPU/Mem และ Peripherals
- ขั้วกลาง I/O กับ Computer (ติดต่อ CPU ผ่าน bus)

I/O Module Function

Control/Status Register --> เก็บสถานะ I/O Modules

Control & Timing

CPU & Device Communication

Data Buffering

Error Detection

I/O Transfer Steps

โอนย้ายข้อมูลเข้าออก

ไปอ่าน status register ว่าพร้อมไหม

ถ้า check status แล้วว่าพร้อม CPU ก็ทำการ transfer data

Input --> transfer เข้าคอม

Output --> transfer เข้า I/O

I/O Techniques

technique การโอนย้ายข้อมูลจาก I/O

1. Programmed I/O

- เขียนโปรแกรมโอนย้ายข้อมูล
- ทำงาน 2 ทอด
- โคตรช้า
- โปรแกรมต้อง run ตลอด
- โปรแกรมคุมหมด ไม่มี Interrupt

2. Interrupt Driven I/O

- I/O Module เอา interrupt ของ Module ต่อกับ CPU
- check Interrupt ทุก instruction เมื่อเกิด Interrupt --> เก็บคำสั่งปัจจุบันใน PC แล้วไปทำงานใน Interrupt
- ใช้การส่ง Interrupt เมื่อพร้อม
- ดีกว่า Programmed I/O แต่ใช้ Hardware เพิ่ม
- มี priority ของการ Interrupt อยู่

3. Direct Memory Access (DMA)

- Programmed กับ Interrupted ต้องใช้ CPU สั่งการ
- เข้าถึงโดยตรงจาก I/O
- มี Hardware เพิ่ม
- CPU บอกข้อมูลให้ DMA แล้วเปิด DMA Controller ให้ทำงาน
- DMA Controller ทำการ transfer นั้น แล้ว CPU ไปทำงานอื่นต่อ
- DMA Controller ส่ง interrupted ไปบอก CPU ว่าเสร็จแล้ว

DMA จะเข้าควบคุม bus ได้ โดยแบ่งใช้จาก CPU แบบแบ่งการใช้งานเร็วๆ หรือตอนที่ไม่จำเป็นต้องใช้ Bus

ใช้สัญญาณ Bus Request ให้ CP (HRQ), ถ้าควบคุมได้จะส่ง Bus ACK (HLDA)

DMA Fly By --> คอยตรงระหว่าง I/O กับ Memory

Cache Controller ต้องรองรับ DMA Controller ด้วย ไม่กันข้อมูลใน Memory กับ Cache ไม่ตรง

I/O Operations

จะรู้ได้ไง I/O แต่อะไรคืออุปกรณ์ไหน

- ใช้ I/O Address
- Address นี้เรียกอีกชื่อได้ว่า Port Number
- ทำได้ด้วยการ Mapping

I/O Mapping

Memory Mapped I/O --> นิยมใช้และออกแบบง่าย

- Device กับ Memory ใช้พื้นที่ร่วมกัน
- I/O จะใช้สัญญาณควบคุมเดียวกับ Memory read/write
- ระวังสับสน และ ความเร็วช้า
- ไม่มี I/O special Command

Isolated I/O

- แยกพื้นที่ Address ของ Memory กับ Device
- แยกสัญญาณควบคุมออกมาเป็น I/O read/write
- มี I/O Special Command

I/O Interface connect I/O Modules to Computer System