

CPU의 구성 요소

- 산술논리 연산장치, ALU ... 연산 장치.
- 제어장치
- 레지스터. → **저장할 모든 데이터는 레지스터를 거쳐야 함.**

명령어, Instruction.

Page 18, Material.

레지스터

범용 레지스터 AX, BX, CX, DX. 16bit. → AH, AL, BH, BL ...
 EAX, EBX, ECX, EDX. 32bit.
 RAX, RBX, RCX, RDX. 64bit.

+ RB-R15. (64bit only) → ABCD 뒤에 따라옴.

다음 장들 때문에...

Src, Dst 인덱스 SI, DI.

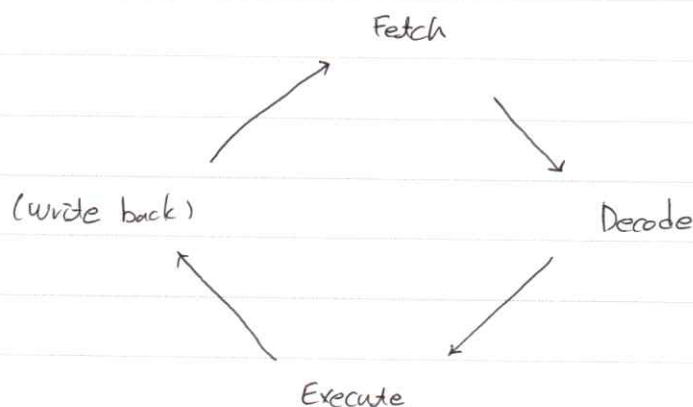
Base Pointer BP

Stack Pointer SP

Instruction Pointer IP → 다음에 실행할 코드의 주소.

Flags (연산 결과). Flag registers. → 연산? 음수?, 9비트 코어?

Instruction Cycle



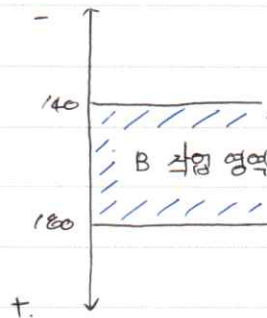
* Page 20-21, Material.

메모리 보호. "CPU는 눈앞에 데이터 처리할 뿐"

Page 24, Material.

- 메모리 보호. 어떤 작업이 다른 작업 영역 침범.
 - 실행 방해. 데이터 손상.
 - ↳ 운영체제 영역 침범 등..

· **Segmentation**. Segmentation 리지스터가 어긋남.



장계 리지스터 140.
한계 리지스터 40.

↓
벗어나면 메모리 오류 인터럽트 발생.
⇒ 대부분 강제 종료. (**Segmentation fault**).

부팅 "운영체제... 메모리에 등록... 부트스트래퍼의 부트스트래핑."

· 하드웨어 점검 → **BIOS** (저장장치 위치) 프로그램 실행.

- **MBR**: Master Boot Record
- **GPT**: GUID Partition Table

"디스크 첫번째 섹터"

전원 → BIOS ROM 실행

(부트스트래퍼) → 무지성 실행... "특수한 영역"

(BIOS 부팅 순서 설정대로 이 링크가 실행)

i.e. 어떤 디스크 어딘가에... 512 Byte.

↓
(MBR) OS 저장위치 보관.
"OS는 여기 있어요!"

"CPU와 하드웨어의 통신 방법"

CPU가 작업을 수행하는 중에 외부에서 들어오는 작업 요청을 처리하는 방법.

f	Polling	"우편함"	CPU가 직접 체크.
	Interrupt.	"택배"	CPU에게 알림.

Polling (aka Programmed IO). • CPU가 주체로 바.

• 머신은 CPU가 Polling 때까지 기다려야 함.

• 주기를 높이면 CPU 부하 ↑.

• Polling Rate "핑 주기"

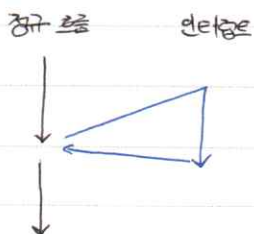
Interrupt (aka Interrupt Driven IO)

• IRQ (Interrupt Request) 전송.

• CPU 활용을 높일 수 있다.

• 지나치게 많은 발생 -- 부하 ↑ (인터럽트 처리 비용 ↑).

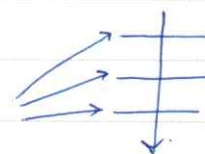
× fn call의 복잡함.



(Nested Interrupt)



(Interrupt Vector)



인터럽트의 구현.

인터럽트.

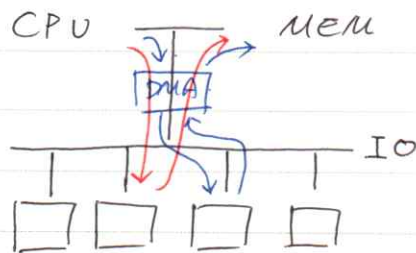
· 하드웨어 인터럽트.

↔ 소프트웨어 인터럽트.

DMA, Direct Memory Access.

"DMA 칩셋이 박혀있는거예요"

· CPU가 입출력에 많은 비용 절감 대책.



* 이전

* 이후

· DMA 칩셋이 CPU에 위주받아
MEM에 3드.

· CPU는 DMA 일하는 중에 다른 일
처리 가능.

· 3드 완료 시 IRQ.

· 동작 모드.

· Cycle Stealing Mode

DMA가 CPU가 동작에 바스 사용하고자 할 때,
바스 CPU가 DMA에게 바스 양보.

· Burst Mode.

DMA가 일을 다할때까지 바스 독점.

· Demand Transfer Mode.

요청 시에만 바스 사용 (DMA).

병렬처리.

동시에 여러 명령 처리 \Rightarrow 성능 높임.

* 동시성과 병렬성.

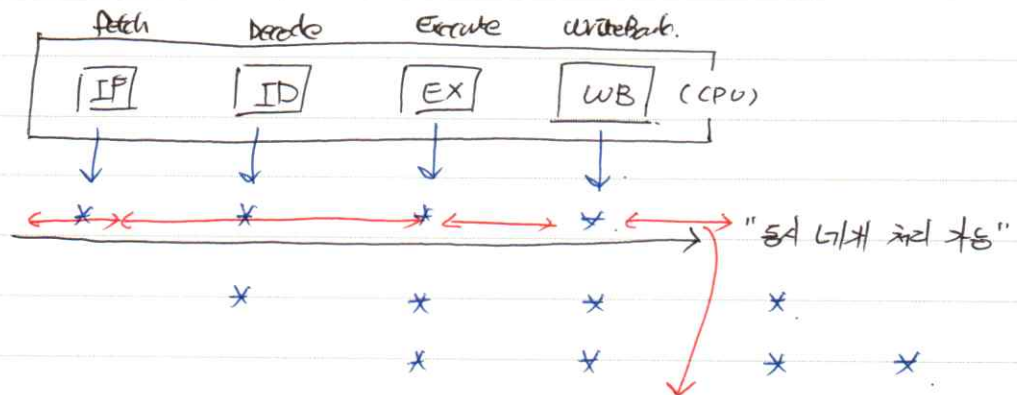
• 동시성, Concurrency.

논리적, 싱글 코어에서 멀티스레딩.

• 병렬성, Parallelism.

물리적, 멀티 코어에서 멀티스레딩.

파이프라이닝.



• 처리할 명령이 길이가 균일해야 함. (예를 들어 길면 처리 지연 발생).

• 처리할 명령이 계속 있어야 함 (파이프라인에 빈 명령이 없어야 함).
 "동료 있는 stage"
 파이프라인의 바늘.
 ↓
 일부러 아무것도 없는 명령을 넣어주는 바늘 처리.

• 명령이 1인 의존성 x.

• 전체 작업 시간 얼마나 줄 수 있?
 $U \uparrow$: 동시 처리할 태스크 $\uparrow \Rightarrow$ 성능 \uparrow .
 Fetch $\uparrow \Rightarrow$ 성능 \downarrow .

Pipeline Hazard.

· 설계로는 처리 한, 단계 끊음 X.

· structural 자원 충돌... 여러 명령 동시 수행시
(같은 자원 사용해야 한다면).

· Control 순차적으로 처리될 경우... "문자에 의해"
 다 등으로 처리 X하게 될 경우.
 ⇒ 바림.

· Data 어떤 명령이 다른 명령에 의존.
 다 수행된 명령 결과값 참조 시도. } → 다 늦어 이럼...

다양한 병렬처리.

· Super Scalar. (=멀티 코어).

· Super-pipeline (=Intel Hyper Threading).

· Super pipelined super scalar.

· VLIW · SW적으로 병렬처리.
 · 어렵고 효율 ↓. } → 잘 안됨.
 · 컴파일러가 잘 해줘야 함

· SIMD. · 하나의 명령어, 여러 데이터 한 번에 처리.