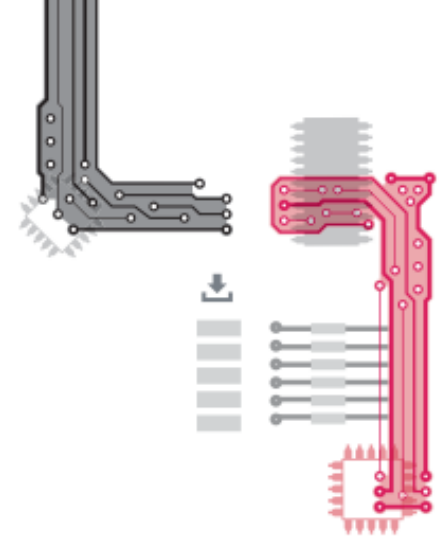


IT CookBook, 컴퓨터 아키텍처

[강의교안 이용 안내]

- 본 강의교안의 저작권은 한빛아카데미(주)에 있습니다.
- 이 자료를 무단으로 전제하거나 배포할 경우 저작권법 136조에 의거하여 최고 5년 이하의 징역 또는 5천만원 이하의 벌금에 처할 수 있고 이를 병과(併科)할 수도 있습니다.



컴퓨터 아키텍처

컴퓨터 구조 및 동작 원리 우종정 지음

Chapter 02. 컴퓨터의 발전과 성능

01. 컴퓨터의 세대별 발전

■ 컴퓨터의 세대별 특징

컴퓨터의 세대	주요 소자	기억장치	특징
0세대 컴퓨터	기계적 릴레이	금속 기어, 카드	-
1세대 컴퓨터 (~1950년대 후반)	진공관	자기드럼	하드웨어 개발에 중점을 두었으며, 부피가 크고 전력 소모가 많음
2세대 컴퓨터 (~1960년대 초반)	트랜지스터	자기코어	고급 언어 등장
3세대 컴퓨터 (~1970년대 중반)	SSI, MSI	RAM, ROM	운영체제 개발, 다중 프로그래밍, 고급언어 활용
4세대 컴퓨터 (~2000년대)	LSI, VLSI	SRAM, DRAM	객체지향언어 사용, 개인용 컴퓨터 등장, 병렬 처리 기능 강화
5세대 컴퓨터 (2000년대 중반~)	VLSI, ULSI, GLSI, SoC	SDRAM, DRAM, flash	비노이만 구조 추구, 멀티코어 사용, 유비쿼터스 컴퓨팅 환경 조성

02. 현대 컴퓨터 발전의 지표

■ 개요

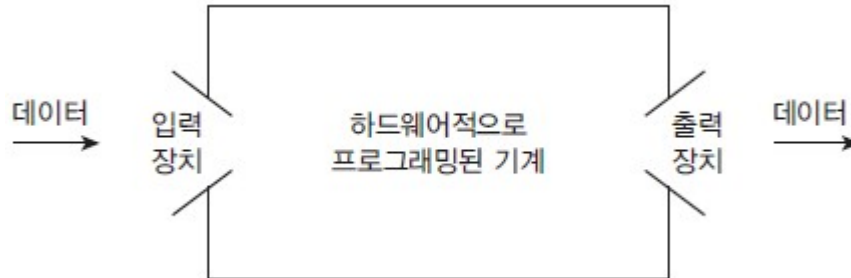
- 디지털 컴퓨터의 추상적인 모델을 제시한 앨런 튜링을 비롯한 수많은 과학자에 의하여 현대 컴퓨터 탄생
- 홀러리스의 천공 카드 시스템, 컴퓨터 마우스의 더글라스 엥겔바트 등
- 폰노이먼 아키텍처, 무어의 법칙 등

■ 폰노이먼 모델

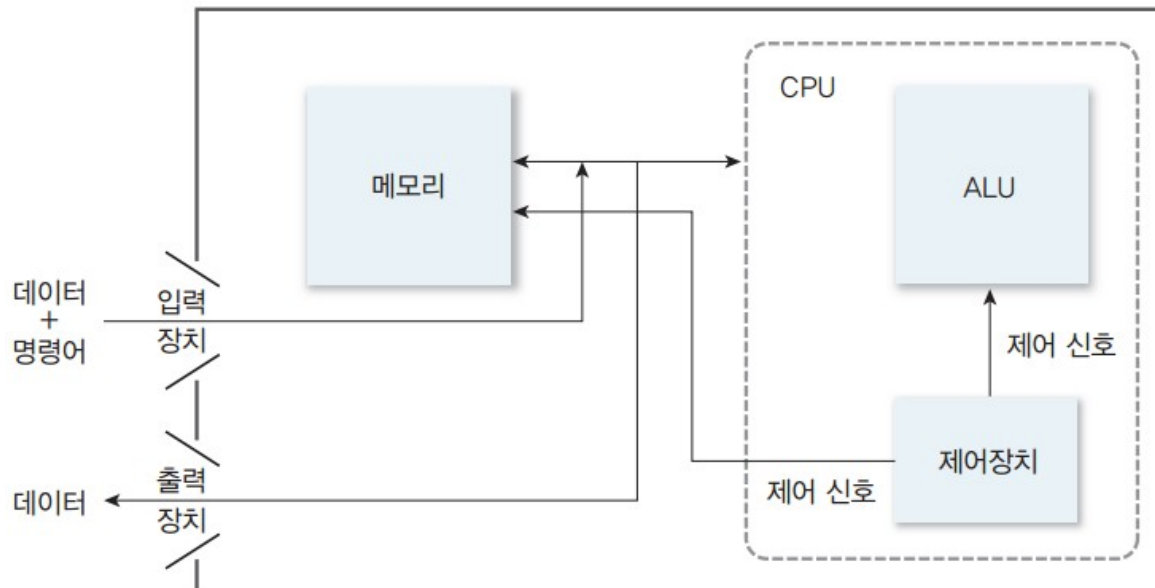
- 프로그램을 저장하고 변경할 수 있는 프로그램 내장식 컴퓨터 모델
- 프린스턴 아키텍처라고도 불림
- 앨런 튜링의 유니버설 기계 모델을 구체화한 형태
- 폰노이만 모델 이전에는 고정결선식 프로그램 컴퓨터(hardwired program computer) 방식을 사용
- 하버드 아키텍처는 다음과 같이 수정한 구조
 - 메모리를 두 개로 분할하여 명령어와 데이터를 별도의 메모리에 저장
 - CPU의 제어장치와 연산장치를 별도의 버스로 각각 연결하여 명령어와 데이터를 병렬로 인출

02. 현대 컴퓨터 발전의 지표

■ 고정결선식 프로그램 컴퓨터



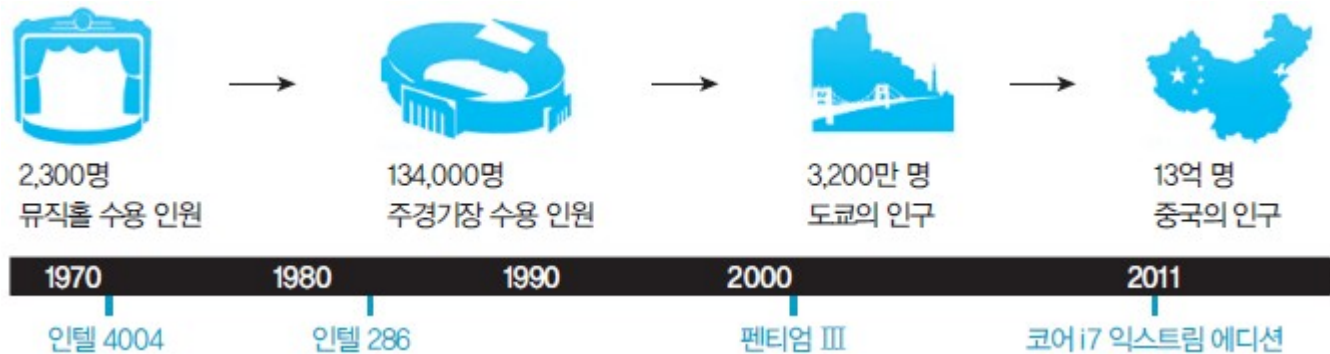
■ 프로그램 내장식 컴퓨터



02. 현대 컴퓨터 발전의 지표

■ 무어의 법칙

- 경험적 관찰에 의해 예측한 것으로, 단일 마이크로칩에 포함된 트랜지스터의 수가 18개월마다 약 2배씩 증가한다는 것을 의미



03. 컴퓨터의 성능

■ 성능의 정의

- 두 종류의 커피 머신 MonoFast와 DoubleSlow
- MonoFast는 한 번에 커피 한 잔을 5분 만에 뽑아내지만 DoubleSlow는 한 번에 커피 두 잔을 7분 만에 뽑아낸다면 둘 중 어느 것의 성능이 더 좋을까?
- 응답시간과 처리율
- CPU 실행시간
 - 프로그램을 수행하는 데 순수하게 사용된 시간
 - 입출력이나 다른 프로그램의 실행 시간이 포함되지 않는다.
- 성능

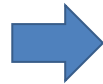
$$\text{성능} = \frac{1}{\text{CPU 실행 시간}}$$

03. 컴퓨터의 성능

■ 암달의 법칙

- 시스템의 일부분을 개선하는 경우 전체 시스템에서 얻을 수 있는 최대 성능 향상을 구할 때 사용
- 주로 병렬화에 의한 시스템 성능 향상의 한계를 지적하는 데 사용
- 시스템 병렬화와 같은 방법으로 도모한 성능 향상은 시스템의 순차적 부분에 필요한 시간에 의해 제한

$$\begin{aligned}s &= \frac{M_2 \text{의 성능}}{M_1 \text{의 성능}} = \frac{M_1 \text{의 실행 시간}}{M_2 \text{의 실행 시간}} \\&= \frac{m}{m \left\{ \frac{f}{n} + (1-f) \right\}} \\&= \frac{1}{\frac{f}{n} + (1-f)}\end{aligned}$$



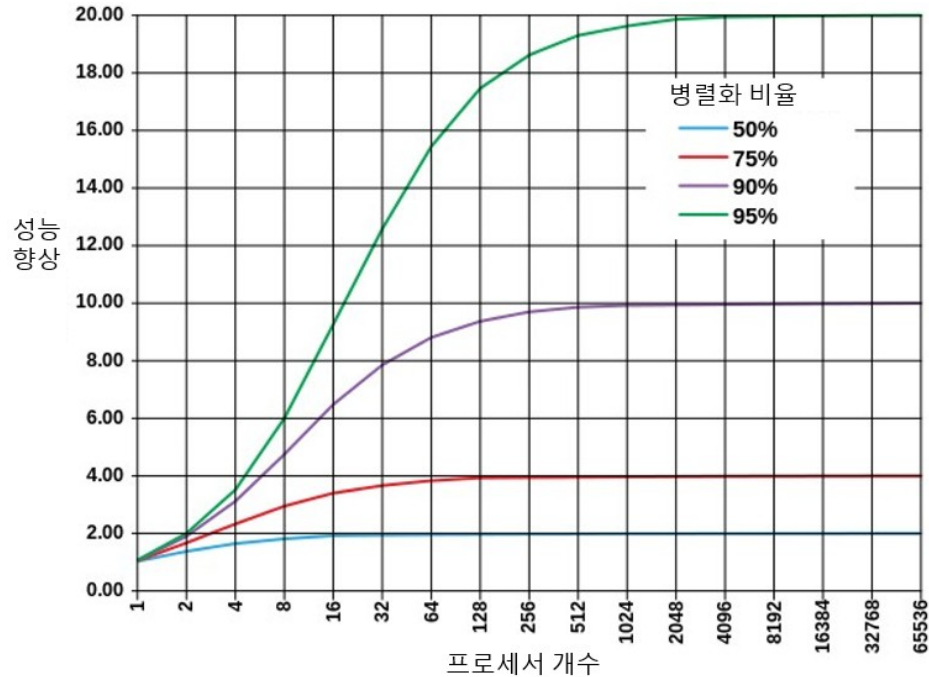
$$s = \frac{1}{\frac{f}{n} + (1-f)} \leq \frac{1}{1-f}$$

$$\lim_{n \rightarrow \infty} s = \lim_{n \rightarrow \infty} \frac{1}{\frac{f}{n} + (1-f)} = \frac{1}{1-f}$$

03. 컴퓨터의 성능

■ 예제 : 암달의 법칙

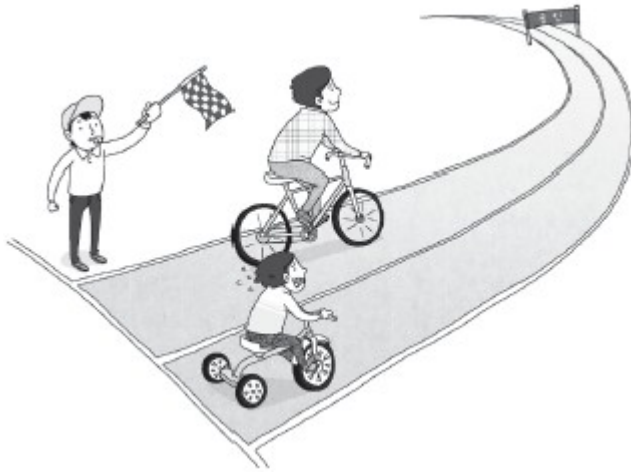
- 하나의 프로그램이 싱글 코어 프로세서를 사용하면 20시간 소요된다.
- 1시간 소요되는 부분은 병렬화할 수 없고, 나머지 19시간 소요되는 부분은 병렬화할 수 있다.
- 병렬화된 부분을 처리하는 데 사용되는 프로세서의 수에는 제한이 없다.



04. CPU 성능 분석

■ 성능에 영향을 미치는 요소

- CPU의 성능을 평가할 때 클럭 속도가 중요
- 하지만, 클럭 속도가 3.0GHz인 CPU가 2.0GHz인 CPU보다 반드시 성능이 우수한 것은 아니다.



04. CPU 성능 분석

■ CPU 실행시간의 분해

$$\frac{\text{CPU 실행 시간}}{\text{프로그램}} = \frac{\text{명령어 개수}}{\text{프로그램}} \times \frac{\text{사이클 개수}}{\text{명령어}} \times \frac{\text{시간}}{\text{사이클}}$$

$$\begin{aligned}\text{프로그램의 CPU 실행 시간} &= \text{명령어 개수} \times \text{평균 CPI} \times \text{사이클 시간} \\ &= \frac{\text{명령어 개수} \times \text{평균 CPI}}{\text{클록 속도}}\end{aligned}$$

03. 컴퓨터의 성능

■ 예제 : 성능에 영향을 미치는 요소

- 세 가지 유형의 명령어 그룹이 있는데, 명령어 유형별 사용 빈도와 CPI는 다음 표와 같다.

명령어	M ₁		M ₂	
	사용 빈도	CPI	사용 빈도	CPI
산술·논리	50%	1	40%	1
적재·저장	30%	2	30%	1
분기	20%	3	30%	4

- 컴퓨터 M1, M2의 클럭 속도는 각각 3GHz, 4GHz이다.
- 두 컴퓨터에서 동일한 개수의 동일한 명령어를 포함하는 특정 프로그램을 수행할 때 어느 컴퓨터가 더 빨리 실행하는가?

04. CPU 성능 분석

■ 성능에 영향을 미치는 요소의 분석

구분	명령어 개수	CPI	클럭 속도
알고리즘	O		
명령어 집합 구조	O	O	Δ
컴퓨터 구현		O	O
컴퓨터 실현			O

05. 벤치마크와 성능 척도

■ 벤치마크의 의미

- 성능을 평가하기 위하여 작업 부하로 선택된 프로그램의 모음
- SPEC 벤치마크
 - SPEC에서 개발한 워크스테이션이나 서버 등의 성능을 평가하는 데 널리 사용되는 벤치마크
 - 예 : 계산 집약적인 작업 부하를 위한 SPEC CPU2006, 메일 서버를 위한 SPECmail2009, 웹 서버를 위한 SPECweb2009 등

■ 평가 척도

- CPU를 위한 CPI, 클럭 속도, MIPS, MFLOPS
- CPU와 메모리 시스템 등의 성능을 종합적으로 평가할 수 있는 성능 척도인 SPECratio
- 기호 조작 및 논리 추론 능력에 중점을 둔 인공지능 컴퓨터의 성능 평가 척도인 등
- 그러나 어떤 성능 척도도 MIPS와 MFLOPS처럼 컴퓨터 시스템의 공급자에 의해 왜곡되어 사용될 수 있음

05. 벤치마크와 성능 척도

■ MIPS

- 단위 시간, 즉 1초에 수행한 명령어 개수를 100만으로 나눈 값

$$\text{MIPS} = \frac{\text{명령어 개수}}{\text{실행 시간} \times 10^6} = \frac{\text{클럭 속도}}{\text{CPI} \times 10^6}$$

■ MFLOPS

- 1초에 실행한 부동소수점 연산의 수를 100만으로 나눈 값

$$\text{MFLOPS} = \frac{\text{부동 소수점 연산 개수}}{\text{실행 시간} \times 10^6}$$

05. 벤치마크와 성능 척도

■ 예제 : MIPS와 성능

- 다음과 같은 명령어 배합을 가진 프로그램

연산 종류	비율	CPI	연산 종류	비율	CPI
산술·논리	40%	1	저장	12%	2
적재	20%	2	분기	28%	2

- 최적화 컴파일러를 사용하면 산술 및 논리 연산이 절반으로 줄어들고
- CPU의 클럭률은 1.0GHz라고 가정하자
- 이때 최적화 컴파일러를 사용하는 경우와 사용하지 않는 경우의 MIPS 값을 비교하라.

감사합니다.

컴퓨터 아키텍처

컴퓨터 구조 및 동작 원리 우종정 지음

