마이크로프로세서 개요

2017-S

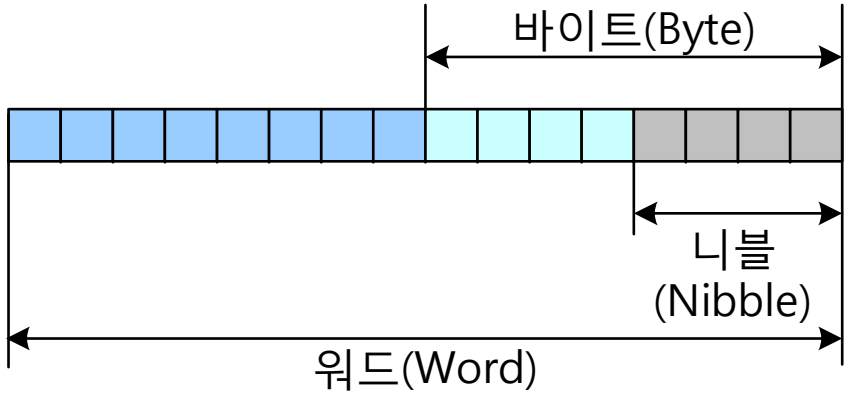
마이크로프로세서설계실험

디지털 시스템의 단계적 분류

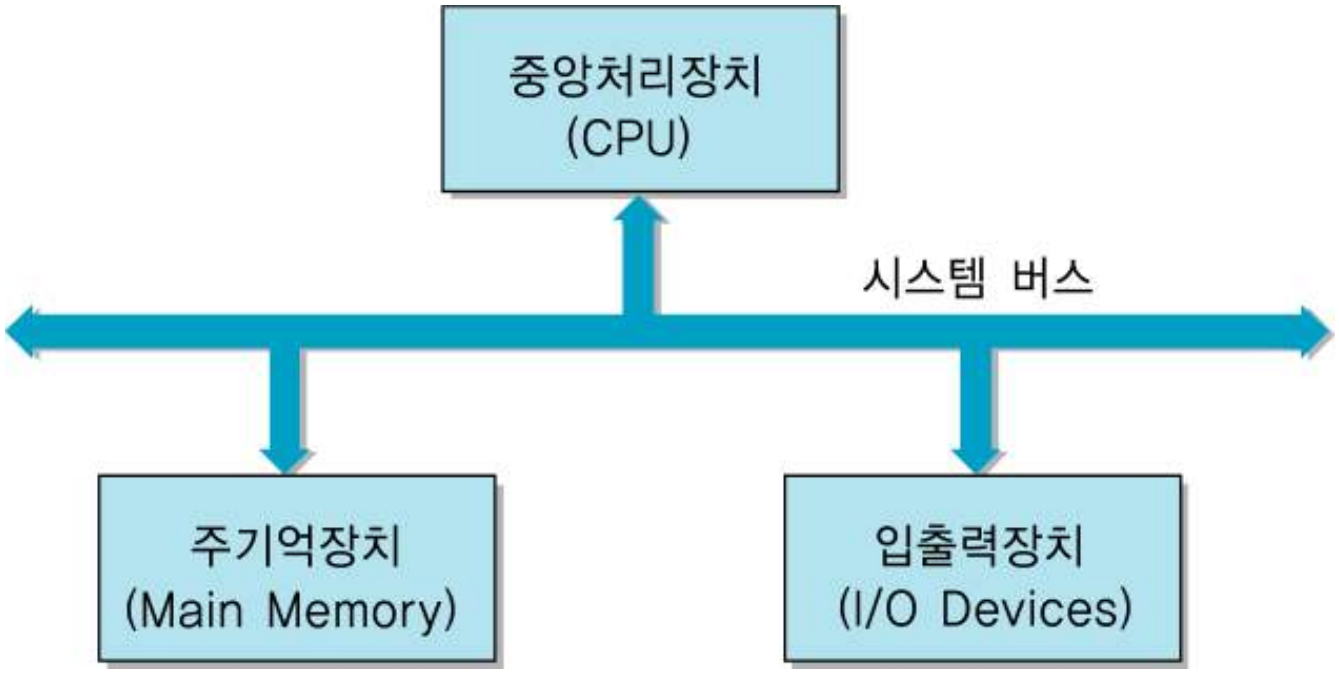
	내 용	예
1	부품 레벨	트랜지스터, 다이오드, 저항, 콘덴서 등
2	소규모 집적회로: 소규모 기능별 논리회로	기본 논리 게이트 등 (NOT, AND, NAND, EX-OR 등)
3	중·대규모 집적회로: 중규모 기능별 논리회로	덧셈기, 계수기, 멀티플렉서 등
4	고집적 회로: 다수의 기능을 가진 논리회로	마이크로프로세서 등
5	시스템: 다수의 IC로 구성된 복잡한 시스템	컴퓨터 등

기본 용어

- 2진수(Binary)
  - High level(5V) : 1, Low level(0V) : 0
- 비트(Bit)
  - 2 진수의 1 자리수
- 니블(Nibble)
  - 2 진수의 4 자리수
- 바이트(Byte)
  - 2 진수의 8 자리수
- 워드(Word)
  - 대부분, 2 진수의 16 자리수 (PIC 명령어 ➔ 14 자리가 1 Word)
- 더블워드(Double Word)
  - 2 진수의 32 자리수
- 16 진수(Hexadecimal)
  - 2진수 4개의 비트 (니블) ➔ 16진수 한 자리
  - 0000 ➔ ’0’, 0001 ➔ ‘1’, 0010 ➔ ’2’ … 1010 ➔ ’A’, 1011 ➔ ’B’, … 1111 ➔ ‘F’
  - ABH ➔ 0ABH
- MSB (Most Significant Bit) / LSB (Least Significant Bit)



컴퓨터의 일반적인 구조

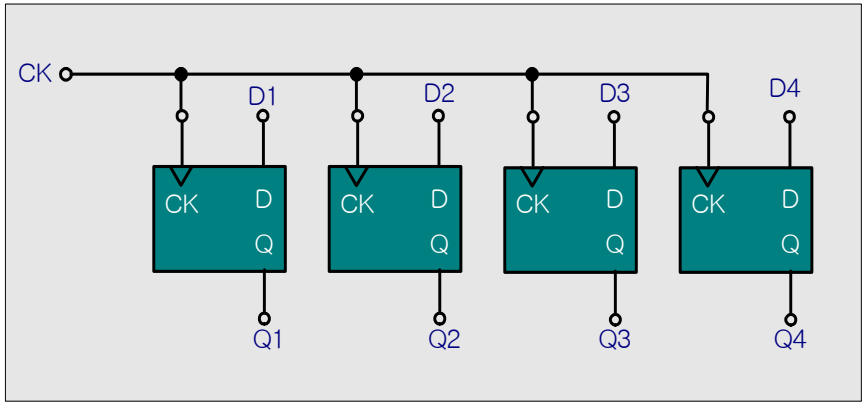


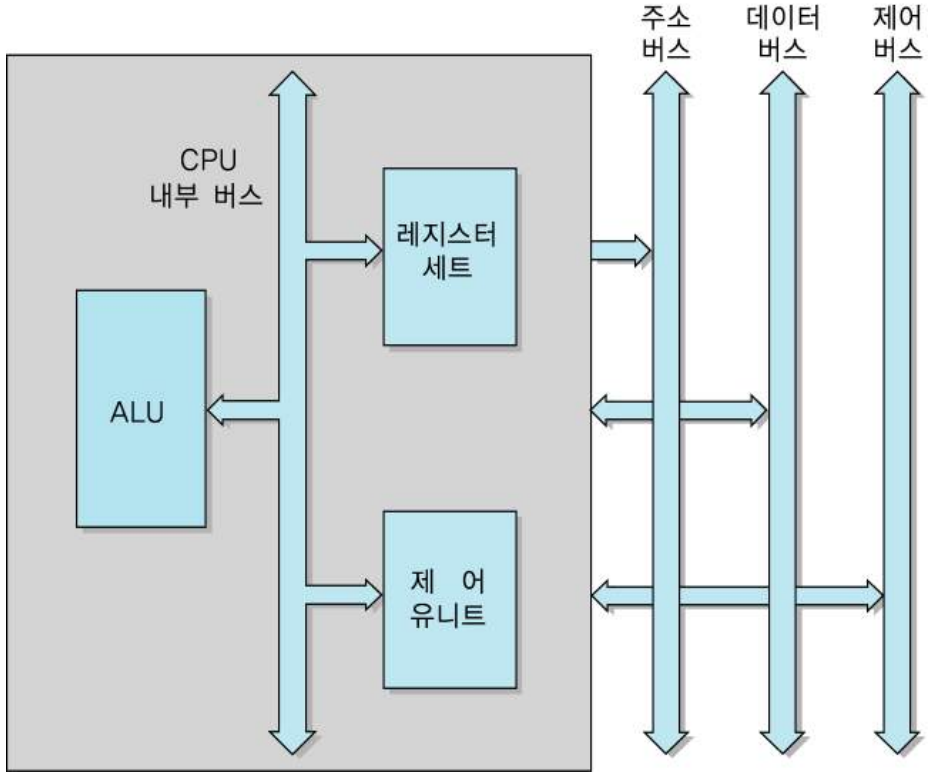
* 버스 (Bus) : CPU와 시스템 내의 다른 요소들 사이에 정보를 교환하는 통로

CPU (Central Processing Unit)의 기본 구조

- 산술논리 연산장치(Arithmetic and Logical Unit: ALU)
- 레지스터 세트(Register Set)
  - 레지스터: CPU 내부 기억공간
  - D Flip-Flop 으로 구성된 기억 소자

ex) 4-bit 레지스터



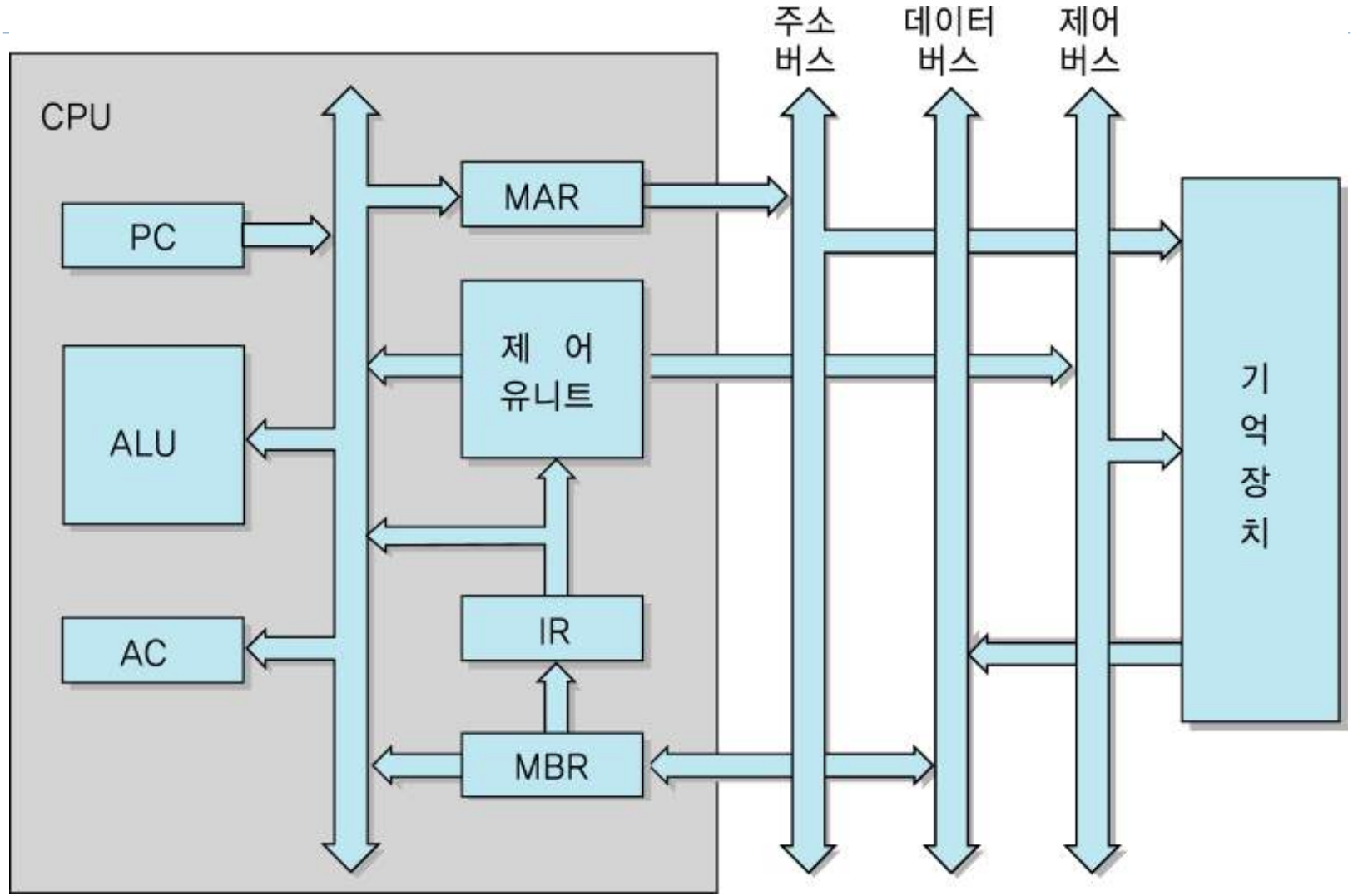


- 제어 장치(Control Unit: CU)

주요 레지스터

- 프로그램 카운터(Program Counter: PC)
  - 다음에 인출할 명령어의 주소를 가지고 있는 레지스터
  - 각 명령어가 인출된 후에는 자동적으로 일정 크기(한 명령어 길이) 만큼 증가
  - 분기(branch) 명령어가 실행되는 경우에는 목적지 주소로 갱신
- 누산기(Accumulator: AC) cf. W-reg in PIC
  - 데이터를 일시적으로 저장하는 레지스터.
  - 레지스터의 크기는 CPU가 한 번에 처리할 수 있는 데이터 비트 수 (단어 길이)
- 명령어 레지스터(Instruction Register: IR)
  - 가장 최근에 인출된 명령어 코드가 저장되어 있는 레지스터

데이터 통로가 표시된 CPU 구조

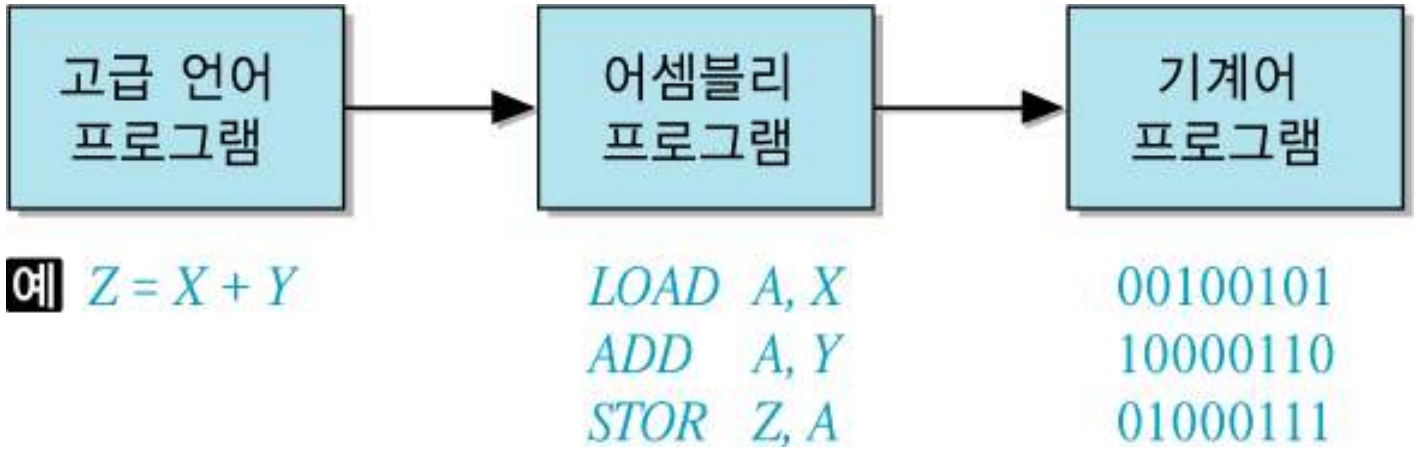


프로그램 코드

- 기계어(machine language)
  - 기계 코드(machine code), 컴퓨터 하드웨어 부품들이 이해할 수 있는 언어
  - 2진수 비트들로 구성
- 어셈블리 언어(assembly language)
  - 어셈블리 코드(assembly code), 고급 언어와 기계어 사이의 중간 언어
  - 저급 언어(low-level language), 기계어와 1:1 대응
- 고급 언어(high-level language)
  - 영문자와 숫자로 구성되어 사람이 이해하기 쉬운 언어
  - C, PASCAL, FORTRAN, COBOL 등

고급 언어 → 어셈블리 언어 → 기계어

- Z = X + Y
  - LOAD A,X : 기억장치 X번지의 내용을 읽어 레지스터 A에 적재(load)
  - ADD A,Y : 기억장치 Y번지 내용을 읽어 레지스터 A에 적재된 값과 더하고 결과를 다시 A에 적재
  - STOR Z,A : 그 값을 기억장치 Z 번지에 저장(store)



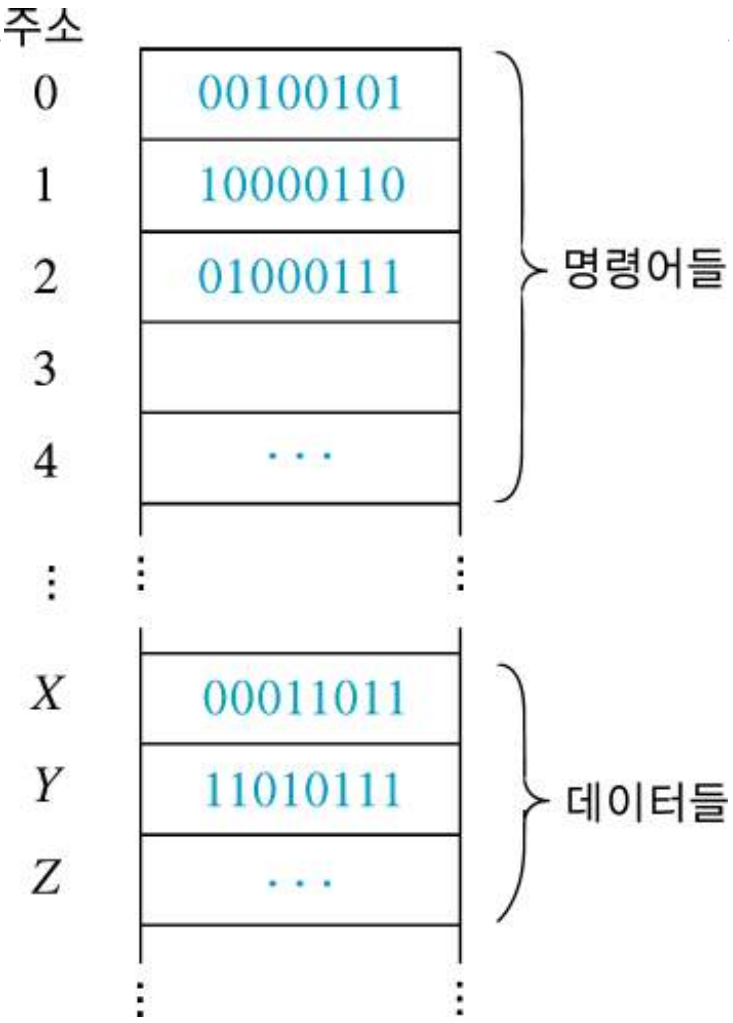
번역 소프트웨어

- 컴파일러(compiler)
  - 고급언어 프로그램을 기계어 프로그램으로 번역하는 소프트웨어
- 어셈블러(assembler)
  - 어셈블리 프로그램을 기계어 프로그램으로 번역하는 소프트웨어
  - 니모닉스(mnemonics)
    - 어셈블리 명령어가 지정하는 동작을 개략적으로 짐작할 수 있도록 하기 위하여 사용된 기호
    - 'LOAD', 'ADD', 'STOR' 등

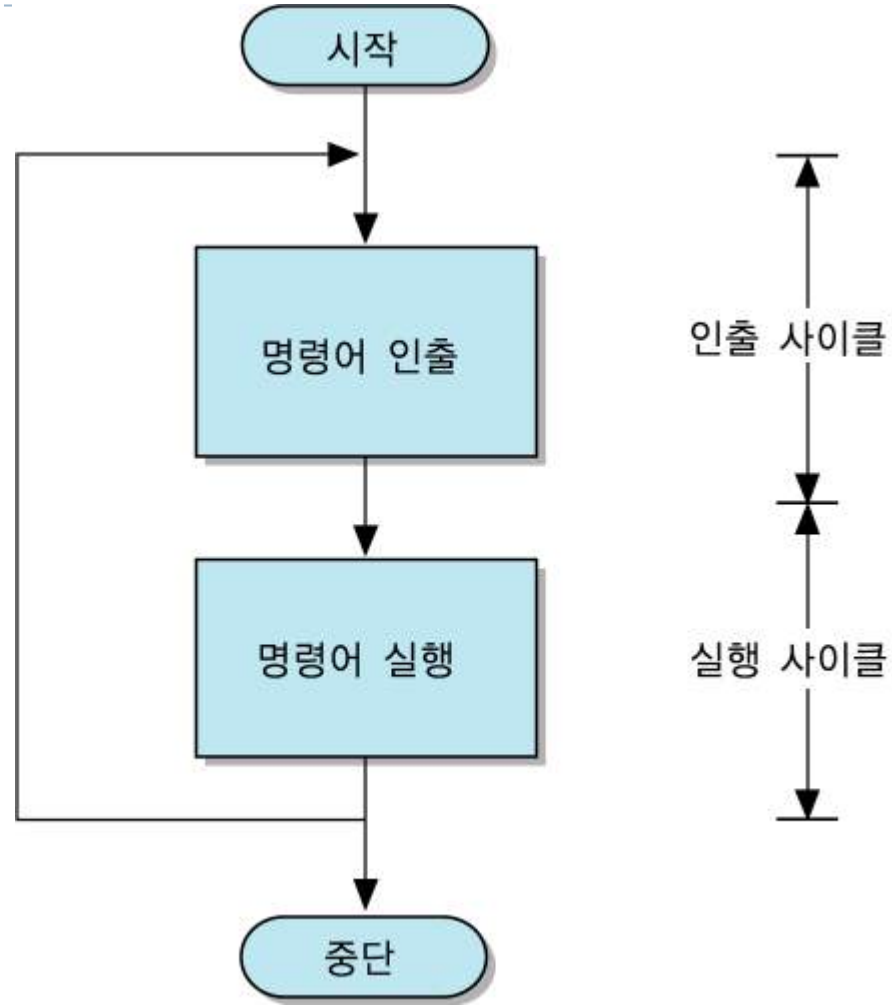
기계어 형식

- 연산 코드(op code)
  - CPU가 수행할 연산을 지정해 주는 비트들
  - 비트 수 = 3이면, 지정할 수 있는 연산의 최대 수는 $2^3 = 8$
- 오퍼랜드(operand)
  - 적재될 데이터가 저장된 기억장치 주소 혹은 연산에 사용될 데이터
  - 비트의 수 = 5이면, 주소 지정할 수 있는 기억장소의 최대 수는 $2^5 = 32$

프로그램과 데이터의 기억장치 저장



기본 명령어 사이클

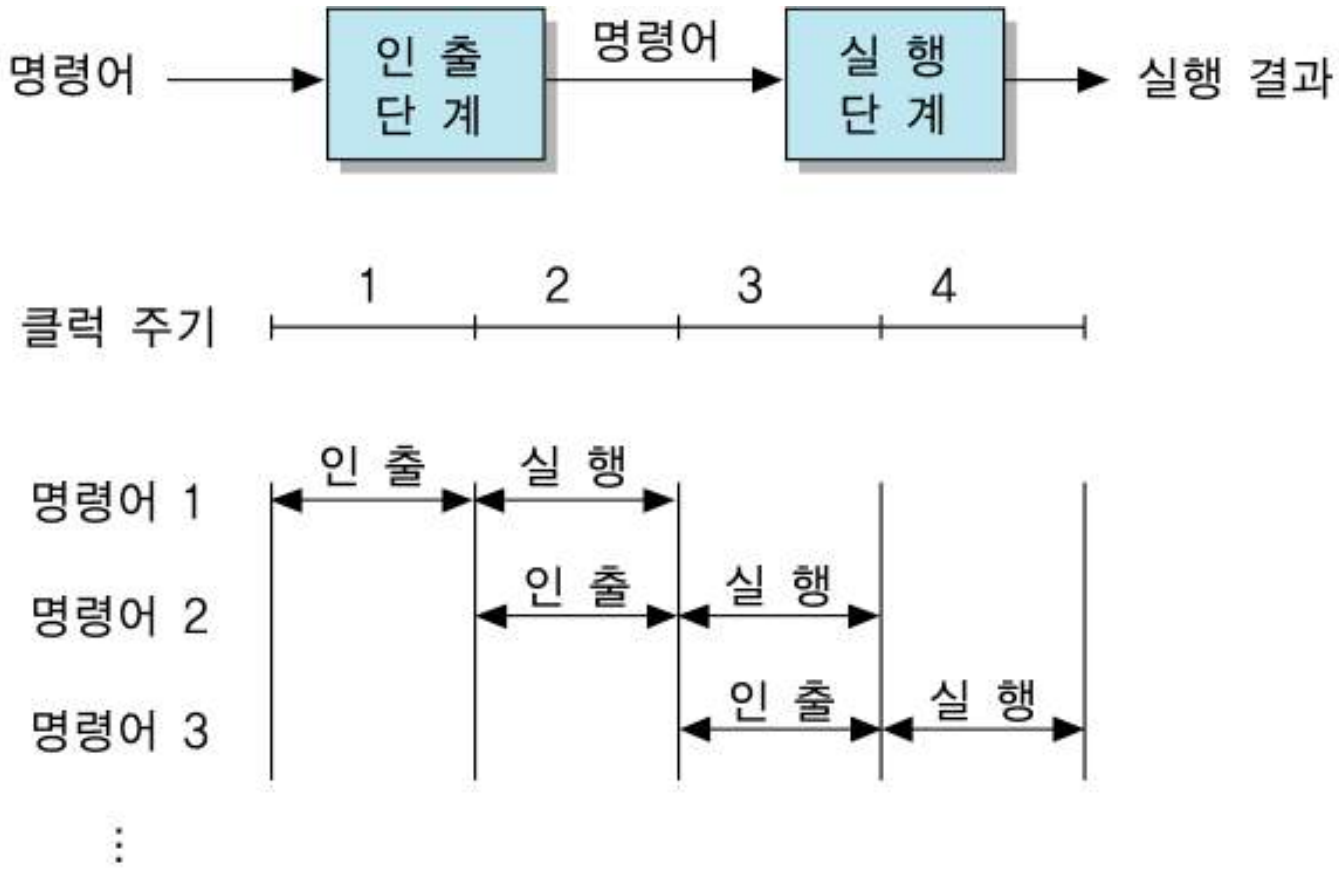


명령어 파이프라이닝 (instruction pipelining)

- CPU의 프로그램 처리 속도를 높이기 위하여 CPU 내부 하드웨어를 여러 단계로 나누어 동시에 처리하는 기술

- 2-단계 명령어 파이프라인(two-stage instruction pipeline)
  - 명령어를 실행하는 하드웨어를 인출 단계(fetch stage)와 실행 단계(execute stage)라는 두 개의 독립적인 파이프라인 모듈들로 분리
  - 두 단계들에 동일한 클럭을 가하여 동작 시간을 일치
    - 첫 번째 클럭 주기에서는 인출 단계가 첫 번째 명령어를 인출
    - 두 번째 클럭 주기에서는 인출된 첫 번째 명령어가 실행 단계로 보내져서 실행되며, 그와 동시에 인출 단계는 두 번째 명령어를 인출

2-단계 명령어 파이프라인과 시간 흐름도



CPU와 기억장치의 접속

- 버스(bus)
  - CPU와 메모리, I/O 소자 등과의 연결에 사용되는 신호선

- 시스템 버스(system bus)
  - 주소 버스(address bus)
  - 데이터 버스(data bus)
  - 제어 버스(control bus)

시스템 버스

- 주소 버스(address bus)
  - CPU가 외부로 발생하는 주소 정보를 전송하는 신호 선들의 집합
  - 주소 선들의 수는 CPU와 접속될 수 있는 최대 기억장치 용량을 결정
    - 주소 버스의 비트 수 = 16 비트라면,
      최대 2^{16} = 64K 개의 기억 장소들의 주소를 지정 가능

- 데이터 버스(data bus)
  - CPU가 기억장치 혹은 I/O 장치와의 사이에 데이터를 전송하기 위한 신호 선들의 집합
  - 데이터 선들의 수는 CPU가 한 번에 전송할 수 있는 비트 수를 결정
    - 데이터 버스 폭 = 32 비트라면,
      CPU와 기억장치 간의 데이터 전송은 한 번에 32 비트씩 가능

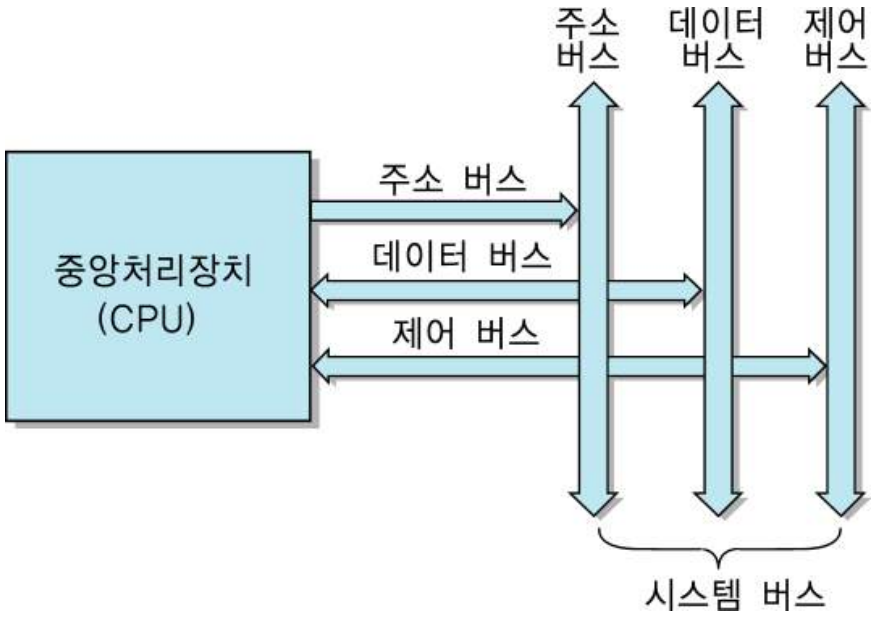
시스템 버스

- 제어 버스(control bus)
  - CPU가 시스템 내의 각종 요소들의 동작을 제어하기 위한 신호 선들의 집합
  - 기억장치 읽기/쓰기(Memory Read/Write) 신호
  - I/O 읽기/쓰기(I/O Read/Write) 신호

CPU와 시스템 버스

- 주소 버스 : 단방향성(uni-directional bus)
  - 주소가 CPU로 부터 기억장치 혹은 I/O 장치들로 전송되는 정보이기 때문

- 데이터 버스, 제어 버스 : 양방향성(bi-directional)
  - 읽기와 쓰기를 모두 해야 하기 때문



CPU와 기억장치 (1/2)

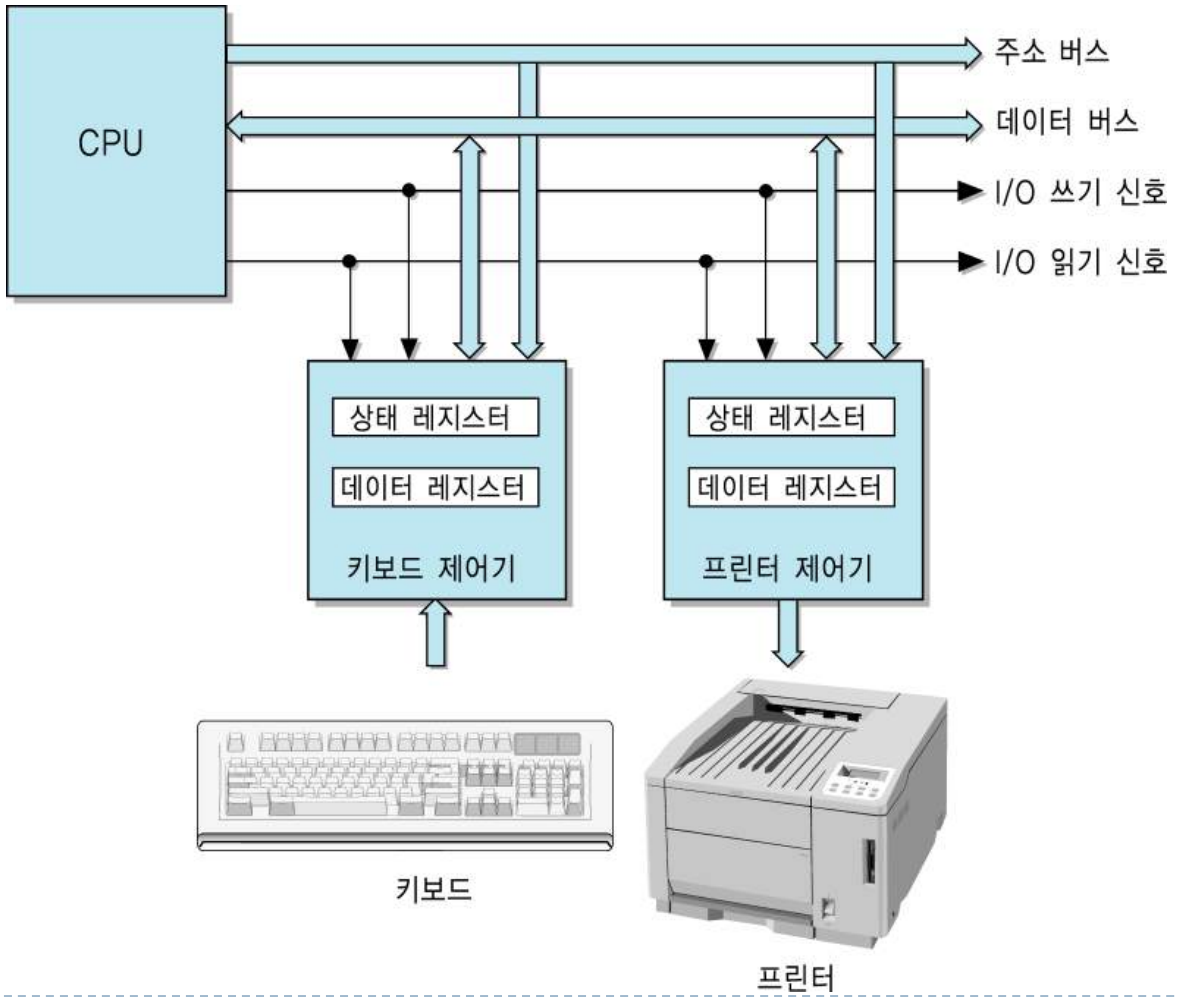
- 기억장치 쓰기 동작
  - CPU가 데이터를 저장할 기억 장소의 주소와 저장할 데이터를 각각 주소 버스와 데이터 버스를 통하여 보내면서 동시에 쓰기 신호를 활성화
  - 기억장치 쓰기 시간(memory write time)
    - CPU가 주소와 데이터를 보낸 순간부터 저장이 완료될 때까지 시간

CPU와 기억장치 (2/2)

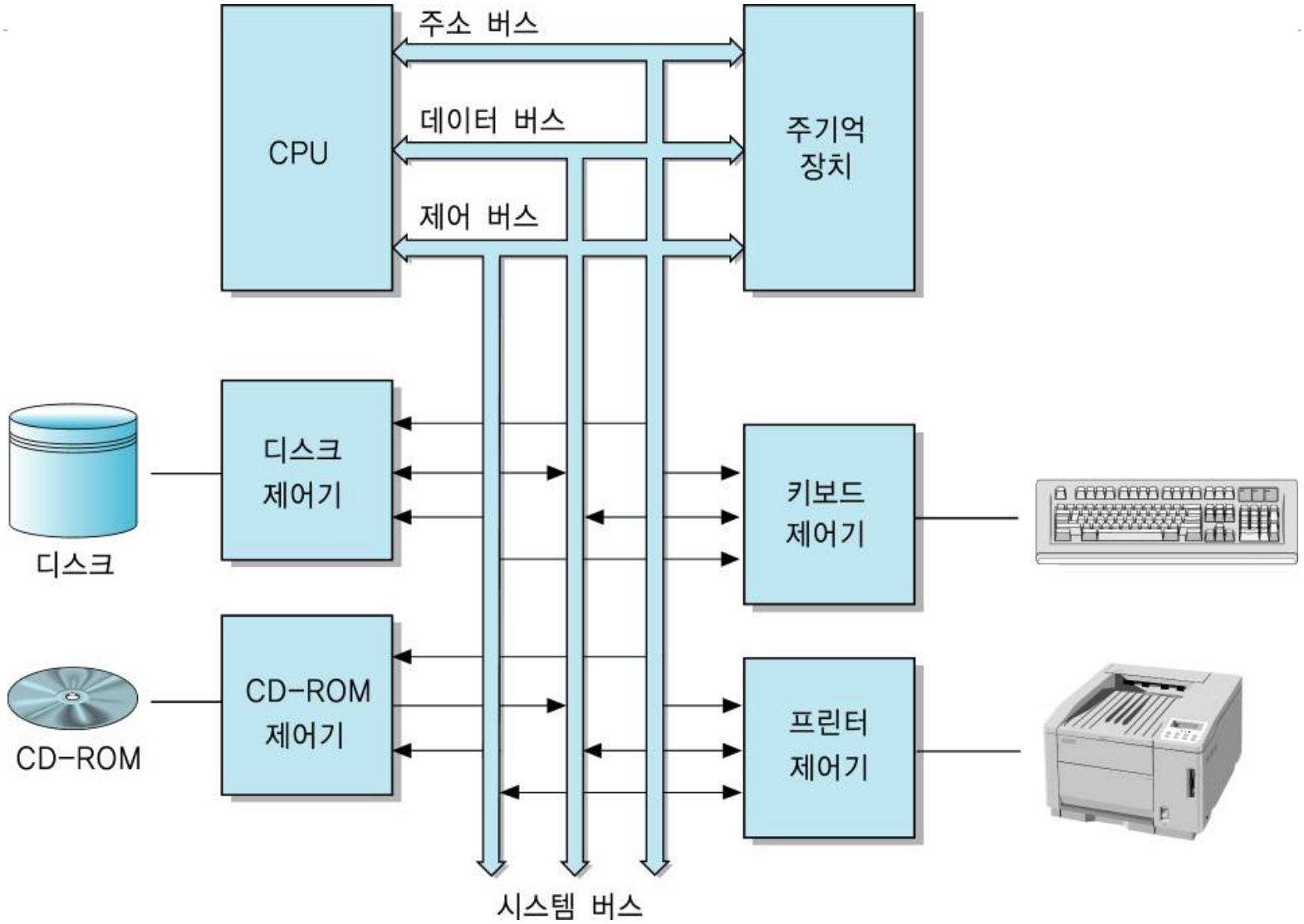
- 기억장치 읽기 동작
  - CPU가 기억장치 주소를 주소 버스를 통하여 보내면서 읽기 신호를 활성화
  - 일정 지연 시간이 경과한 후에 기억장치로부터 읽혀진 데이터가 데이터 버스 상에 실리며, CPU는 그 데이터를 버스 인터페이스 회로를 통하여 읽음
  - 기억장치 읽기 시간(memory read time)
    - 주소를 해독(decode)하는 데 걸리는 시간과 선택된 기억 소자들로부터 데이터를 읽는 데 걸리는 시간을 합한 시간

CPU와 I/O 장치의 접속

- CPU - 시스템 버스 – I/O 장치 제어기 – I/O 장치



컴퓨터시스템의 전체 구성도



CPU Architecture

- 명령어 구조에 따른 구분
  - CISC (Complex Instruction Set Computer)
  - RISC (Reduced Instruction Set Computer)

- 버스구조에 따른 구분
  - Princeton (Von Neumann)
  - HARVARD

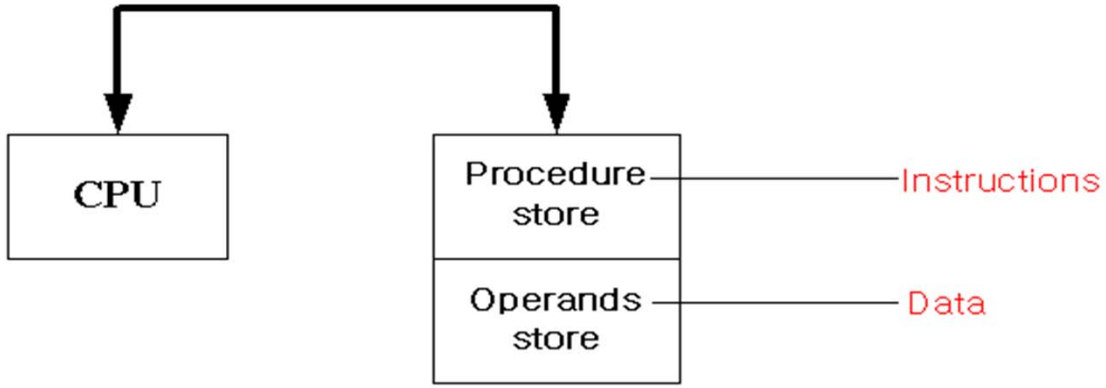
CISC vs. RISC

구분	CISC	RISC
CPU instruction	* 명령어 개수가 많음 * 가변적인 길이 * 실행 사이클도 명령어 마다 다름	* 명령어 개수가 적음 * 명령어 길이 고정적 * 실행 사이클도 모두 동일
회로구성	복잡	단순
메모리 사용	높은 밀도의 명령어 사용으로 메모리 사용이 효율적	낮은 밀도의 명령어 사용으로 메모리 사용이 비효율적
프로그램 측면	명령어를 적게 사용	* 상대적으로 많은 명령어가 필요 * 파이프라이닝
컴파일러	다양한 명령을 사용하므로 컴파일러 복잡	명령어 개수가 적어서 단순한 컴파일러 구현 가능

버스구조에 따른 구분 (1/2)

- Princeton Architecture
  - Common (Shared) Bus and Common Memory for Program (Instructions, Codes) and Operands (Data)

* Bus/Memory are Time-shared between Program/Operand Access



버스구조에 따른 구분 (2/2)

- Harvard Architecture
  - Separate Bus and Separate Memory for Program (Instructions, Codes) and Operands (Data)

* Concurrent access of Program and Operands

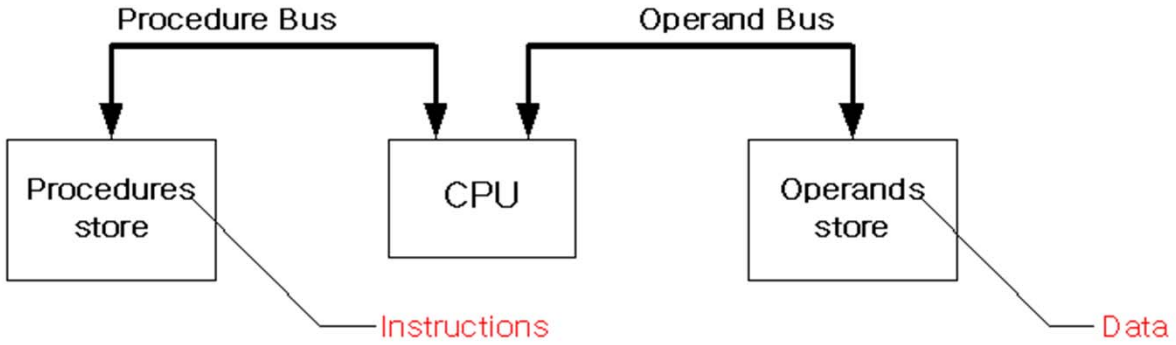


Fig. 1-9 Class-I Harvard Architecture

마이크로프로세서 vs. 마이크로컨트롤러

- 마이크로프로세서(microprocessor)
  - 컴퓨터의 중앙처리장치(Central Processing Unit; CPU)를 단일 IC (Integrated Circuit) 칩에 집적시켜 만든 반도체 소자로서 1971년에 미국의 Intel사에 의하여 세계 최초로 만들어졌으며 오늘날은 이를 흔히 MPU(MicroProcessor Unit)라고 부르기도 함

- 마이크로컨트롤러(microcontroller)
  - 마이크로프로세서 중에 1개의 칩내에 CPU 기능은 물론이고 일정한 용량의 메모리(ROM, RAM 등)와 입출력 제어 인터페이스 회로까지를 내장한 것으로 흔히 MCU (MicroController Unit)라고 부르기도 함
  - 범용의 목적보다는 기기 제어용에 주로 사용되므로 붙여진 이름
  - 1개의 소자만으로 완전한 하나의 컴퓨터 기능을 갖추고 있으므로 "단일 칩 마이크로컴퓨터(one-chip 또는 single-chip microcomputer)"라고도 부름
  - 가장 많이 사용되는 MCU 계열
    - 8051, PIC, AVR

MCU의 장점

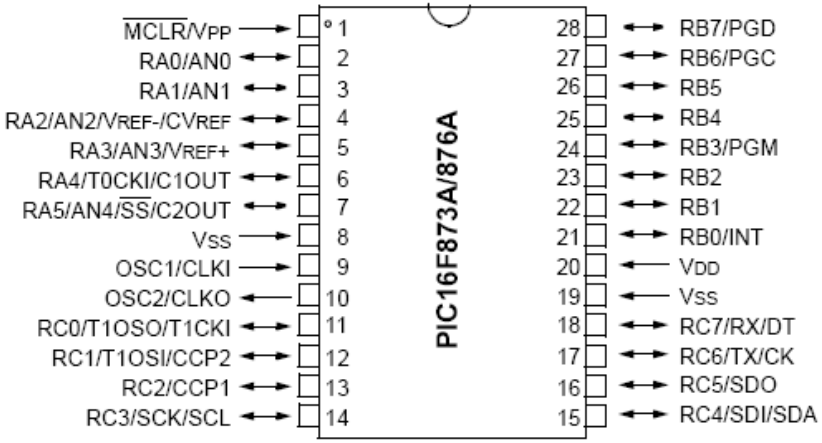
- **제품의 소형경량화**
  - 시스템의 컨트롤러 부분이 MCU와 극히 소수의 외부 소자들도 간단히 구성되므로 크기와 무게가 현저히 줄어들고 소비전력도 적어진다. 이에 따라 부수적으로 전원장치까지도 소형경량화 된다.
- **저가격**
  - 컨트롤러 부분이 단순화됨에 따라 부품비, 제작비, 개발비가 감소되고, 개발기간도 단축된다.
- **시스템의 신뢰성 향상**
  - 컨트롤러가 단순화되어 부품수가 적어지고 신뢰도가 높은 소자를 사용하므로 고장률이 감소하며 유지보수가 용이해진다.
- **시스템의 융통성 증가**
  - 하드웨어에 의존하는 부분을 소프트웨어로 처리할 수 있게 되므로 기능의 변경이나 확장에 보다 유연하게 대응할 수 있다.

마이크로컨트롤러의 응용

- 가전기기
- 통신
- PC 주변기기
- 자동차
- 공장자동화
- 의료기기
- 기타

PIC16F876A 특징 (1/3)

- Pin diagram



- Device features

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I^2C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2

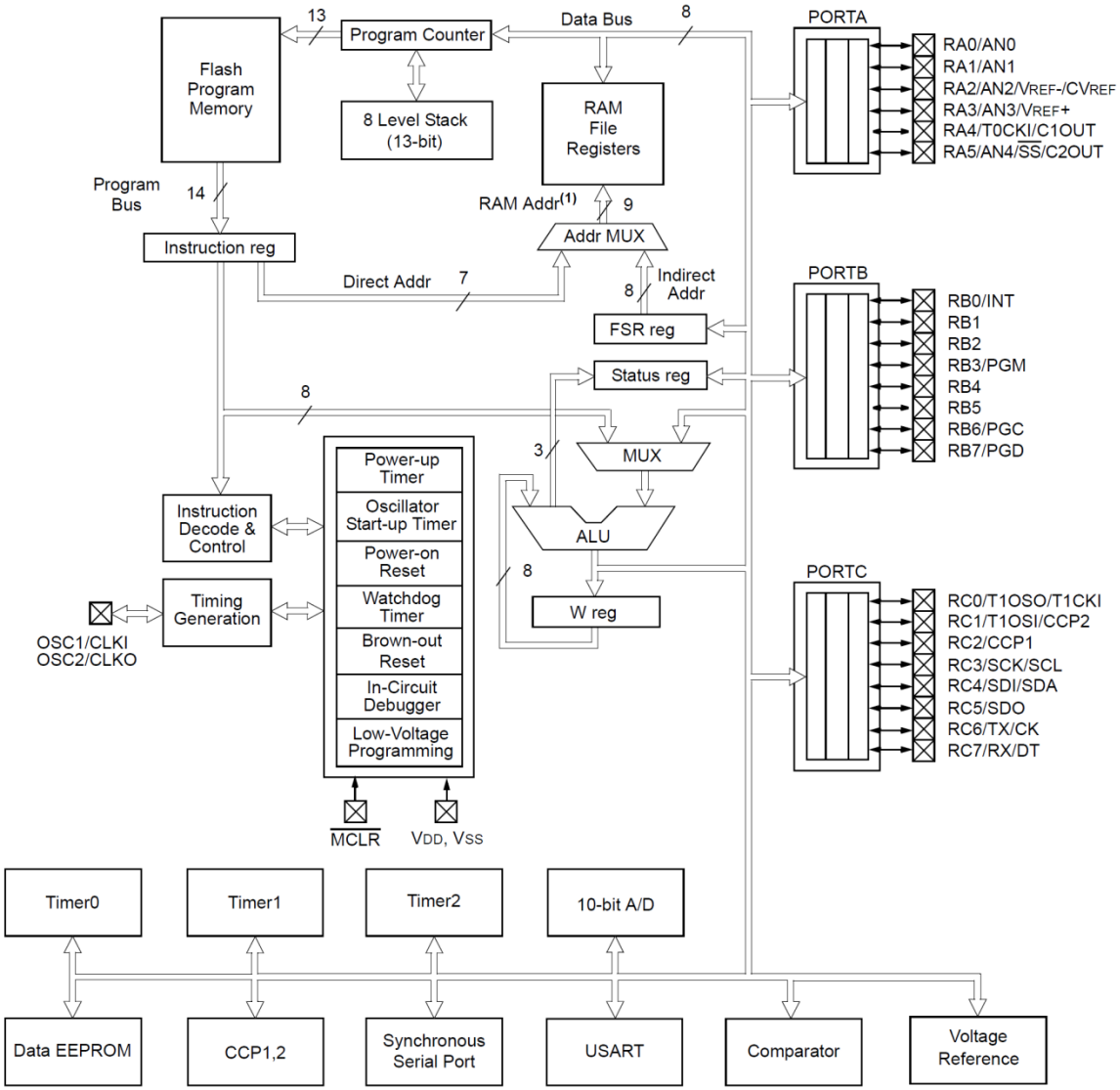
PIC16F876A 특징 (2/3)

TABLE 1-1: PIC16F87XA DEVICE FEATURES

Key Features	PIC16F873A	PIC16F874A	PIC16F876A	PIC16F877A
Operating Frequency	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz
Resets (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
Flash Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory (bytes)	128	128	256	256
Interrupts	14	15	14	15
I/O Ports	Ports A, B, C	Ports A, B, C, D, E	Ports A, B, C	Ports A, B, C, D, E
Timers	3	3	3	3
Capture/Compare/PWM modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Analog Comparators	2	2	2	2
Instruction Set	35 Instructions	35 Instructions	35 Instructions	35 Instructions
Packages	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN

PIC16F876A 특징 (3/3)

- 고 성능의 RISC CPU
- 35개의 명령어
- 모든 명령은 1 사이클(단, 분기가 발생할 경우 2 사이클)
- 동작속도 : DC ~ 20MHz의 클럭 입력 (DC-200nsec 명령 사이클)
- 8Kx14 워드 플래시 프로그램 메모리, 368x8 바이트 데이터 메모리, 256x8 바이트 EEPROM
- 8단계 하드웨어 스택
- 워치독 타이머(WDT : Watchdog Timer)
- 파워 절약을 위한 슬립모드
- 프로그램 보호
- 발진기 옵션 선택 가능
- 저 전력, 고속 CMOS FLASH/EEPROM
- 2핀을 이용한 인 서킷 시리얼 프로그래밍(ICSP : In-Circuit Serial Programming)
- 5V 전원으로 프로그래밍
- 2핀을 이용한 인 서킷 디버깅
- 2.0~5.5V 동작 가능
- 높은 싱크/소스 전류 25mA

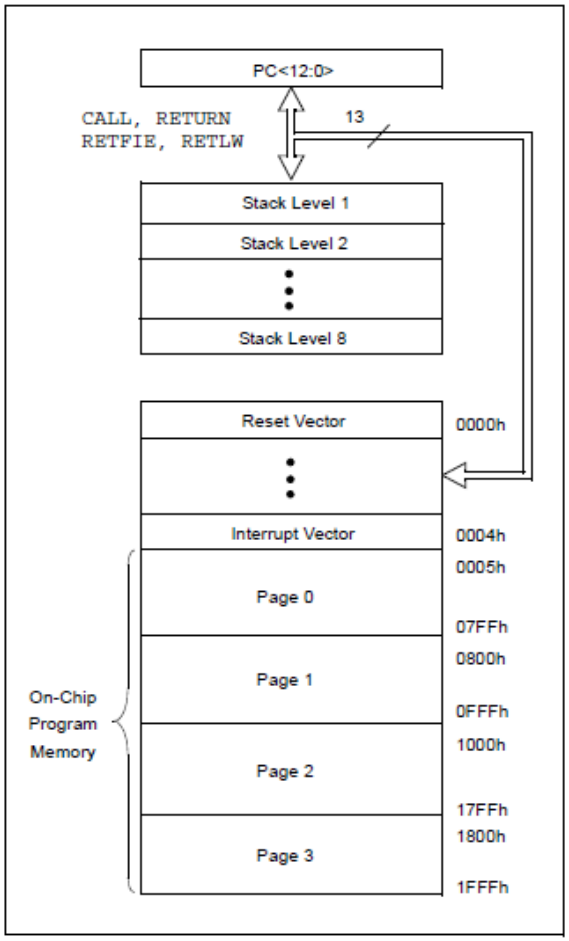


Device	Program Flash	Data Memory	Data EEPROM
PIC16F873A	4K words	192 Bytes	128 Bytes
PIC16F876A	8K words	368 Bytes	256 Bytes

Note 1: Higher order bits are from the Status register.

메모리구조

FIGURE 2-1: PIC16F876A/877A PROGRAM MEMORY MAP AND STACK



2.2 Data Memory Organization

The data memory is partitioned into multiple banks which contain the General Purpose Registers and the Special Function Registers. Bits RP1 (Status<6>) and RP0 (Status<5>) are the bank select bits.

RP1:RP0	Bank
00	0
01	1
10	2
11	3

FIGURE 2-3: PIC16F876A/877A REGISTER FILE MAP

Bank 0	File Address	Bank 1	File Address	Bank 2	File Address	Bank 3	File Address
Indirect addr.(*)	00h	Indirect addr.(*)	80h	Indirect addr.(*)	100h	Indirect addr.(*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD(1)	08h	TRISD(1)	88h		108h		188h
PORTE(1)	09h	TRISE(1)	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved(2)	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved(2)	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h	General Purpose Register 16 Bytes	117h	General Purpose Register 16 Bytes	197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch	CMCON	9Ch		11Ch		19Ch
CCP2CON	1Dh	CVRCON	9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
General Purpose Register 96 Bytes	20h	General Purpose Register 80 Bytes	A0h	General Purpose Register 80 Bytes	120h	General Purpose Register 80 Bytes	1A0h
			EFh		16Fh		1EFh
accesses 70h-7Fh	7Fh	accesses 70h-7Fh	F0h FFh	accesses 70h-7Fh	170h 17Fh	accesses 70h - 7Fh	1F0h 1FFh

Unimplemented data memory locations, read as '0'.
* Not a physical register.

Note 1: These registers are not implemented on the PIC16F876A.
2: These registers are reserved; maintain these registers clear.

Instruction Set Summary

- Three basic categories
  - **Byte-oriented** operations
  - **Bit-oriented** operations
  - **Literal and control** operations

- READ-MODIFY-WRITE operations
  - For example, a “CLRF PORTB” instruction will read PORTB, clear all the data bits, then write the result back to PORTB.

TABLE 15-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1 .
PC	Program Counter
TO	Time-out bit
PD	Power-down bit

FIGURE 15-1: GENERAL FORMAT FOR INSTRUCTIONS

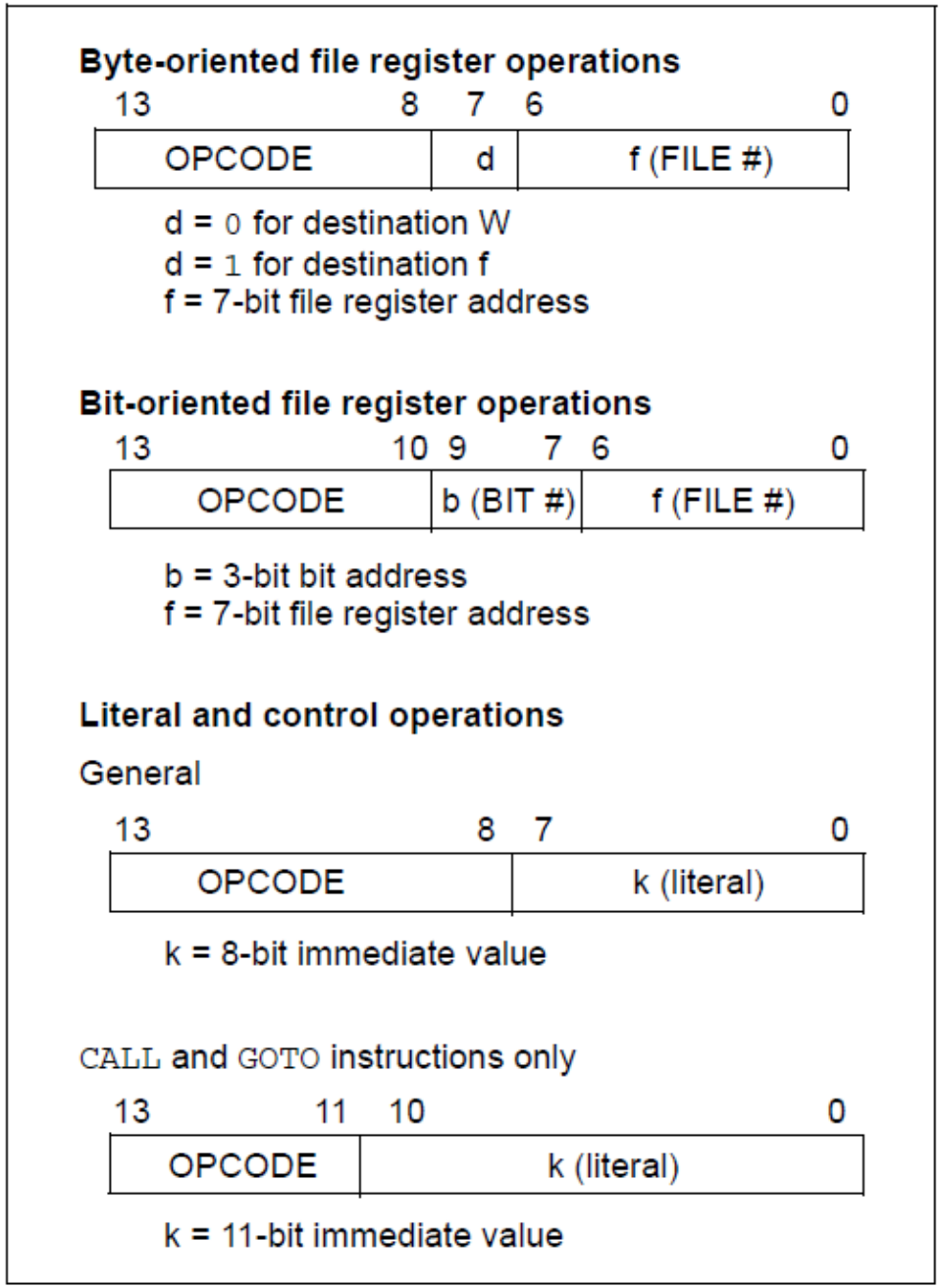


TABLE 15-2: PIC16F87XA INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS						
ADDWF	f, d	Add W and f	1	00 0111 dfff ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00 0101 dfff ffff	Z	1,2
CLRF	f	Clear f	1	00 0001 lfff ffff	Z	2
CLRW	-	Clear W	1	00 0001 0xxx xxxx	Z	
COMF	f, d	Complement f	1	00 1001 dfff ffff	Z	1,2
DECF	f, d	Decrement f	1	00 0011 dfff ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00 1011 dfff ffff		1,2,3
INCF	f, d	Increment f	1	00 1010 dfff ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00 1111 dfff ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00 0100 dfff ffff	Z	1,2
MOVF	f, d	Move f	1	00 1000 dfff ffff	Z	1,2
MOVWF	f	Move W to f	1	00 0000 lfff ffff		
NOP	-	No Operation	1	00 0000 0xx0 0000		
RLF	f, d	Rotate Left f through Carry	1	00 1101 dfff ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00 1100 dfff ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00 0010 dfff ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00 1110 dfff ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00 0110 dfff ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS						
BCF	f, b	Bit Clear f	1	01 00bb bfff ffff		1,2
BSF	f, b	Bit Set f	1	01 01bb bfff ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01 10bb bfff ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01 11bb bfff ffff		3
LITERAL AND CONTROL OPERATIONS						
ADDLW	k	Add Literal and W	1	11 111x kkkk kkkk	C,DC,Z	
ANDLW	k	AND Literal with W	1	11 1001 kkkk kkkk	Z	
CALL	k	Call Subroutine	2	10 0kkk kkkk kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00 0000 0110 0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to Address	2	10 1kkk kkkk kkkk		
IORLW	k	Inclusive OR Literal with W	1	11 1000 kkkk kkkk	Z	
MOVLW	k	Move Literal to W	1	11 00xx kkkk kkkk		
RETFIE	-	Return from Interrupt	2	00 0000 0000 1001		
RETLW	k	Return with Literal in W	2	11 01xx kkkk kkkk		
RETURN	-	Return from Subroutine	2	00 0000 0000 1000		
SLEEP	-	Go into Standby mode	1	00 0000 0110 0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from Literal	1	11 110x kkkk kkkk	C,DC,Z	
XORLW	k	Exclusive OR Literal with W	1	11 1010 kkkk kkkk	Z	