

임베디드시스템 소개 : 개념, 하드웨어

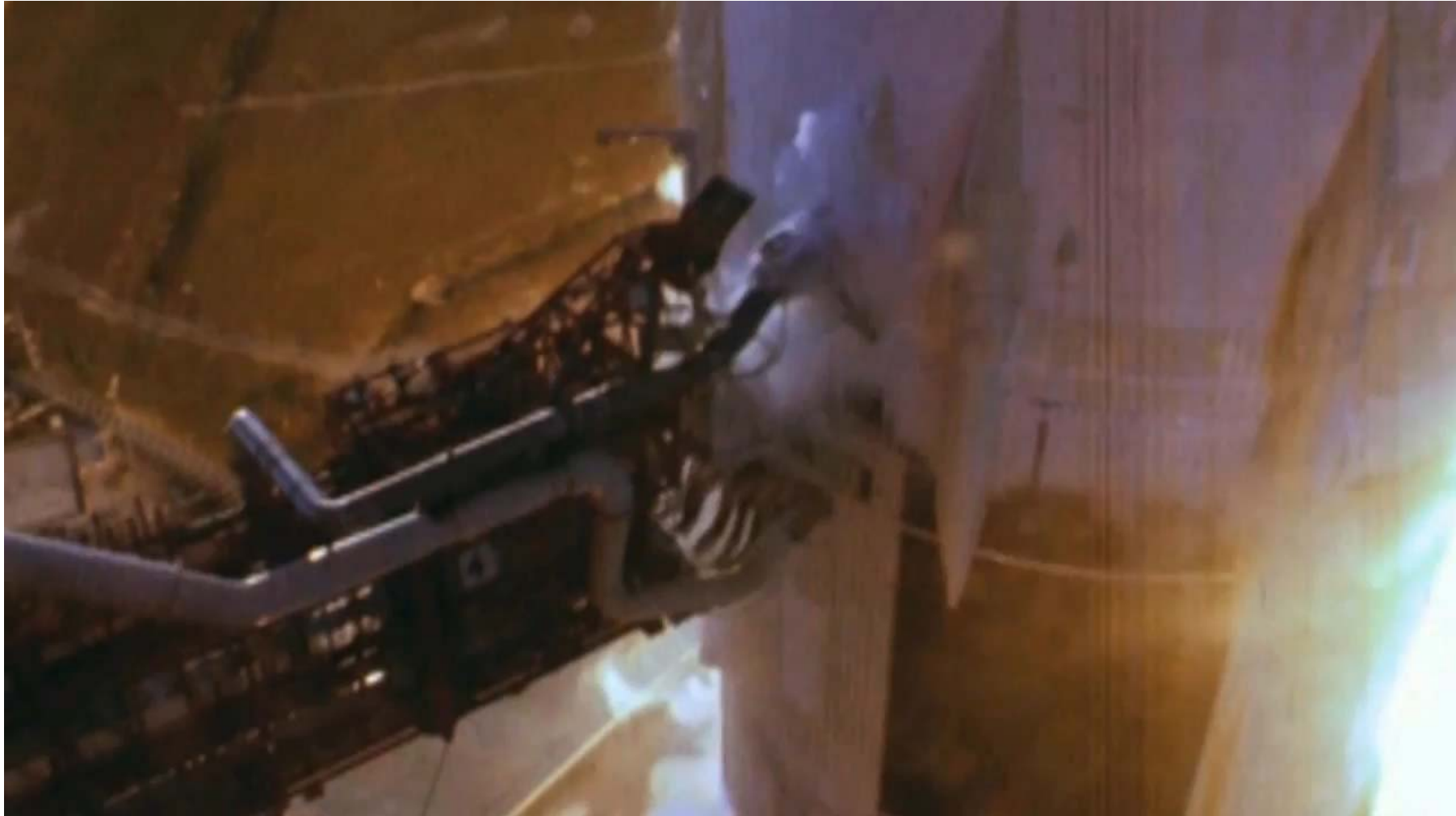


Kim, Eui-Jik

Contents

- 임베디드 시스템
- 임베디드 시스템의 종류
- 임베디드 시스템의 구성요소
- 임베디드 소프트웨어
- 임베디드시스템 소프트웨어의 특징
- 임베디드시스템에서 사용하는 운영체제
- 임베디드 제품 개발 과정
- 하드웨어 기본 용어

What is an Embedded System?



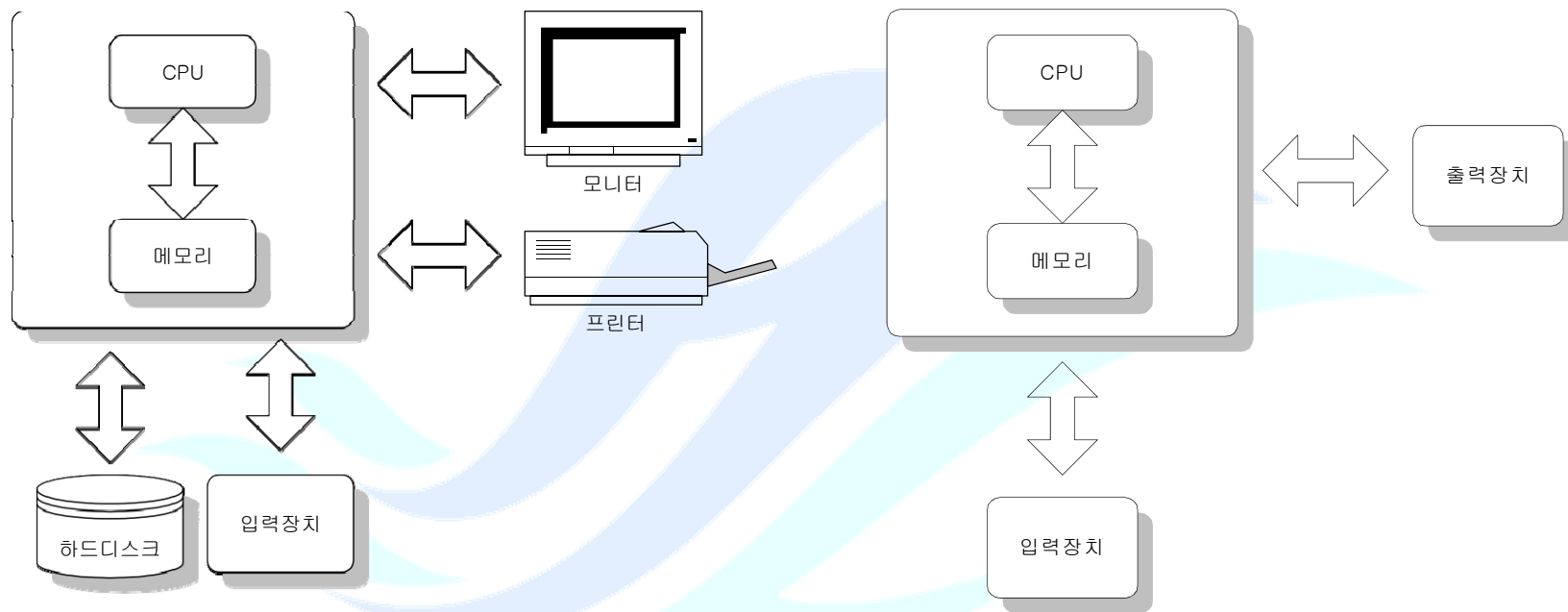
임베디드 시스템

- 임베디드 시스템 (Embedded System)
 - Embedded: '내장되어 있는', '포함되어 있는'
 - 특수한 목적에 대한 기능을 제공하기 위하여 하드웨어와 소프트웨어가 내장된 전자 제어 시스템
 - 사람이 제품의 동작에 개입하지 않으면서 사용자의 요구를 수행하기 위한 시스템
- 임베디드 프로그래밍
 - 임베디드 시스템이 동작하는데 필요한 모든 사항을 제어
- 활용범위
 - 산업용: 자동차, 항공, 군사장비, 의료장비, 공장제어 등
 - 생활용품: 스마트폰, 보안카메라, 도어락, 냉장고, TV 등
 - 사물인터넷을 위한 플랫폼
 - 사물인터넷 (Internet of Things: IoT): 사물에 대한 데이터를 수집하고 제어하며 외부 환경과 연결해 부가서비스를 제공

임베디드 시스템의 종류

- 그래픽 사용자 인터페이스 (Graphic User Interface: GUI)를 위한 입출력 장치의 유무에 따라 구분
- 헤드풀 (Headful)
 - 출력장치와 터치스크린 혹은 키보드 등의 입력장치를 가지고 있는 시스템
 - 사용자의 의도를 입력받아 원하는 결과를 출력
 - ex. ATM, 셋톱박스 (Set-top box), 키오스크
- 헤드리스 (Headless)
 - GUI를 위한 입출력장치가 없음
 - 온도, 습도, 가속도 등의 환경변화를 감지하는 센서를 통한 데이터 입력과 이에 따른 액추에이터 동작 및 알람과 같은 이벤트를 발생시킴
 - ex. IoT의 사물

임베디드 시스템의 구성요소



컴퓨터의 내부 구성

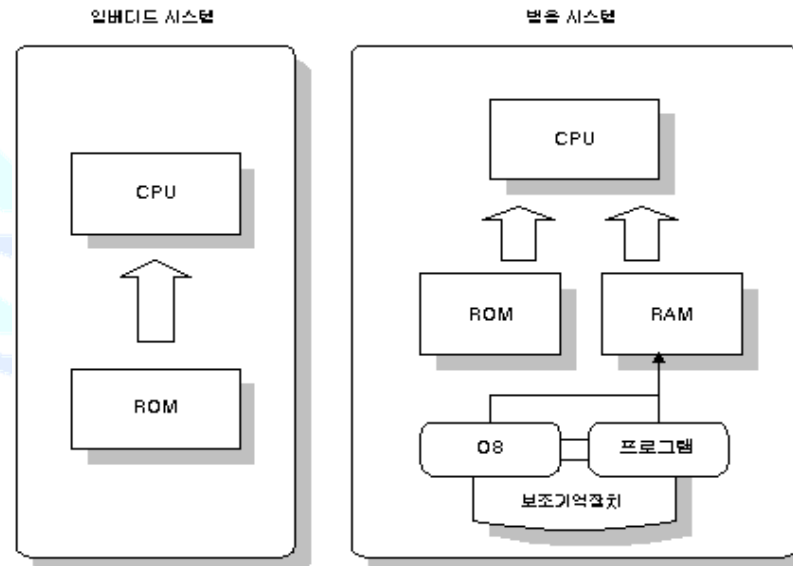
임베디드 시스템의 구성

임베디드 소프트웨어

- 임베디드시스템 소프트웨어는 임의, 단독, 개별적으로 동작하지 않음
 - 임베디드시스템에 사용되는 소프트웨어는 미리 구성되어 있는 하드웨어와 밀접한 관계를 갖고 있기 때문에, 소프트웨어에 작은 변화를 주어도 처리되는 결과에 큰 차이가 나타날 수 있음
- 안정성
 - 의미: 시스템을 얼마나 신뢰할 수 있는가?
 - 임베디드시스템 설계의 가장 중요한 고려사항
 - 주변 상황이 좋지 않은 상태에서도 실행 가능해야 함
 - ex. 기상 및 의료 관련 장비
 - 실시간 문제와도 관련됨

임베디드시스템과 범용시스템의 차이점

- 임베디드 시스템은 특수한 목적을 위해 만들어짐
 - 임베디드시스템
 - 시스템의 프로그램을 ROM이나 플래시메모리에 저장
 - 범용컴퓨터
 - 운영체제나 응용 프로그램을 하드디스크 등의 보조기억장치에 저장했다가 필요한 경우 RAM에 복사해서 실행



임베디드시스템 소프트웨어의 특징

- 독립성
 - 하드웨어와 호환성이 매우 적음
 - HW 호환성 없다고 할 수 있음
 - 하드웨어에 대해 최적화된 프로그램을 만들 수 있는 장점
- 무결성
 - 사람이 관여하지 않고 스스로 제어 하도록 설계
 - 오류없이 완벽한 동작이 이루어지도록 보장되어야 함
- 실시간 제어
 - 시스템을 동작시키는데 정확한 시간 제어가 필요함
- 최적화
 - 시스템 자원에 대한 최적화 필요
 - 프로그램의 구조가 단순해야 함

임베디드시스템에서 사용하는 운영체제

- (1) 운영체제가 없는 임베디드 시스템
 - 소규모 개발 과정에 이용
 - 운영체제가 없는 임베디드 시스템은 해당 기기의 목적에만 충실하도록 만들어짐

```
void main(void)
{
    초기화();
    while(1)
    {
        기능 동작();
    }
}
```

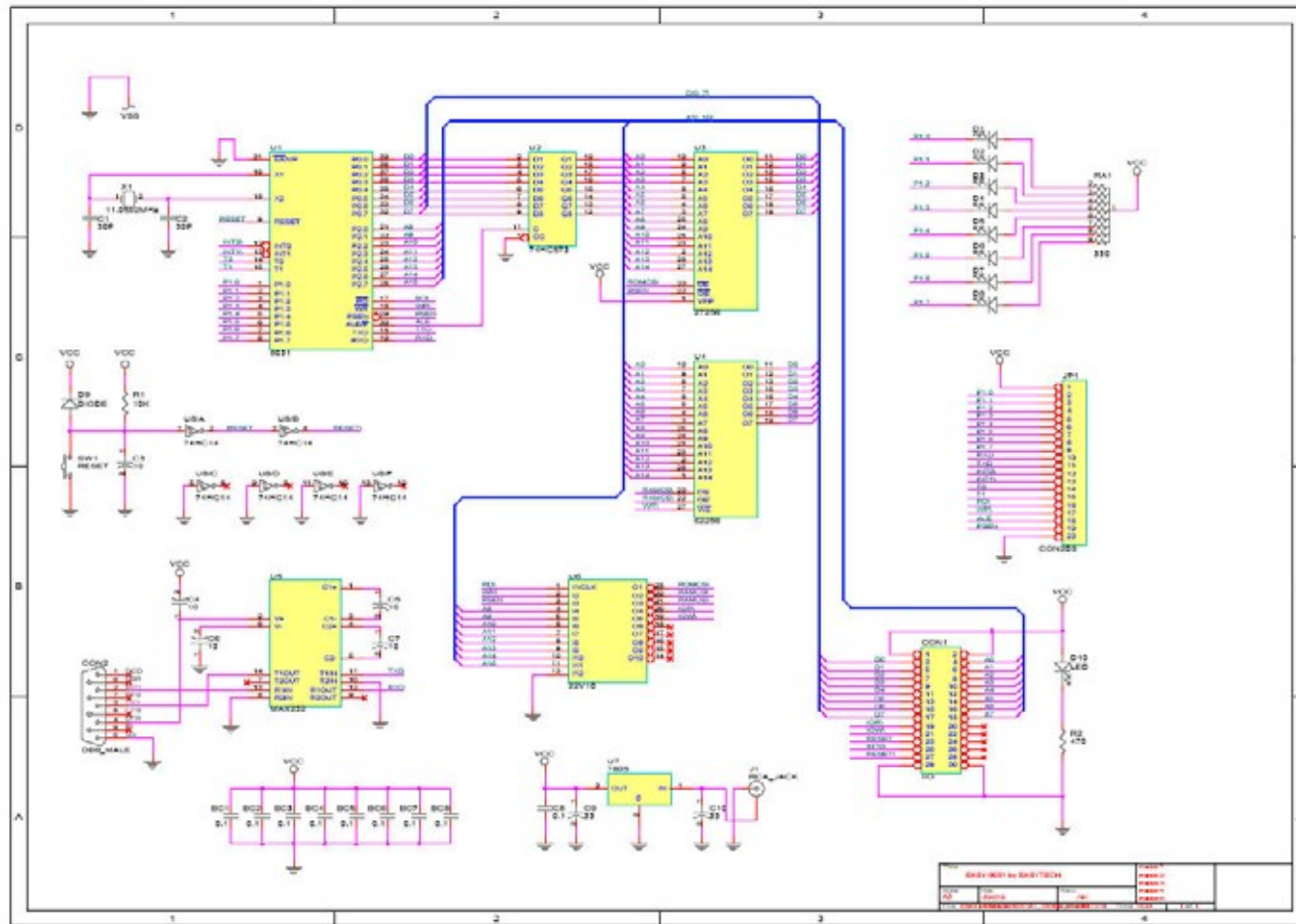
임베디드시스템에서 사용하는 운영체제

- (2) 운영체제가 있는 임베디드 시스템
 - 하나의 제품에 다기능이 탑재된 경우
 - 각각의 기능들을 관리해주는 운영체제가 필수적으로 필요
 - 실시간 운영체제
 - RTOS (Real Time Operating System)
 - 임베디드 시스템에서 처리 시에 필요한 모든 정보나 서비스를 제공하고 처리를 주도하기 때문에 개발자의 입장에서 편리함
 - 비용이 많이 들고, 개발 기간이 오래 걸리는 등의 단점
 - ex. VxWorks, QNX, LynxOS, pSOS, Nucleus 등
 - 임베디드 리눅스
 - 개발 비용이 저렴
 - 모든 소스가 공개되어 있고 공유하는 것이 원칙
 - 하드웨어에 적합한 개발 정보를 얻기 용이
 - 범용적으로 사용 가능한 개발환경 제공

임베디드 제품 개발 과정

- (1) CPU, Memory, 기타 주변 장치(USB, 확장메모리 등) 선정 (HW&SW)
- (2) CAD (Computer Aided Design)로 회로 그리기 (HW)
 - 각종 테스트를 위한 회로용, 테스트보드의 크기는 실제 제품보다 크게 제작
 - OrCAD PSpice 등 사용
- (3) Artwork (HW)
 - CAD에 그려져 있는 것을 실제 부품들에 연결할 수 있도록 데이터(거버 (Gerber)데이터)로 만드는 작업
 - 실제 부품 사이즈, 커넥터 모양, 부품의 위치 선정, 실제 부품들 간의 배선 연결 등의 작업
- (4) PWB (Printed Wiring Board) 제작 (HW)
 - PCB (Printed Circuit Board)
- (5) 부품 장착 (HW)
- (6) 디버깅 (HW&SW)
 - 하드웨어 및 소프트웨어 디버깅
- (7) 회로도 수정
- (8) 양산 테스트
- (9) 품질 검사
- (10) 규격 검사 및 제품 출하

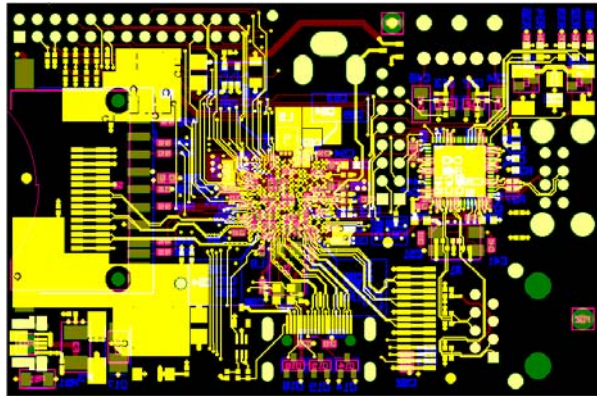
임베디드 제품 개발 과정



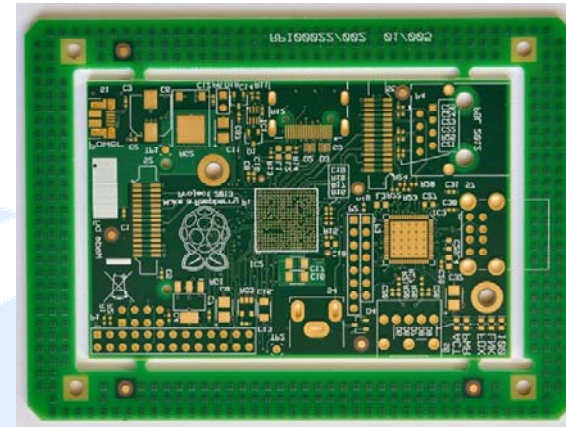
CAD로 그린 회로도

임베디드 제품 개발 과정

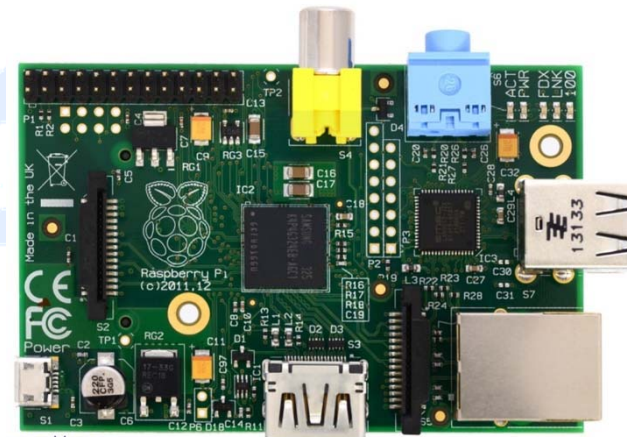
Raspberry Pi – rev 2 B



Artwork 예시

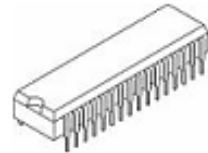


PWB (Printed Wiring Board) 예시



부품 장착

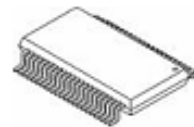
임베디드 제품 개발 과정



Dual In-line Package(DIP)



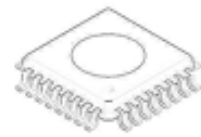
Plastic Leaded Chip Carrier(PLCC)



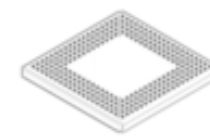
Thin Small Outline Package(TSOP)



Ball Grid Array (BGA)



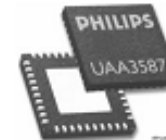
EQAD



Flip Chip Pin Grid Array(FC-PGA)



Plastic Quad Flat Package(PQFP)



Quad Flat Non-leaded package(QFN)

다양한 형태의 칩 패키지

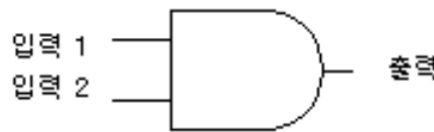
하드웨어 기본 용어

- 대부분의 디지털 회로는 두 종류의 전압 사용
 - 0볼트: **그라운드(GND), V_{SS} , 접지 또는 Low**
 - 3볼트 또는 5볼트: **V_{CC} , V_{DD} 또는 High**
- 회로 안의 신호선은 순간적으로 2종류 중 하나의 전압을 나타냄
- 회로의 동작은 각 부품의 신호선이 Low→High (Rising edge), High→Low (Falling edge)로 바뀌는 것
- 신호선이 데이터나 어드레스를 나타낼 때 Low는 0, High는 1로 표현
- 신호선 종류
 - 마이크로프로세서 리셋
 - Chip Enable: 메모리 칩의 데이터 획득
 - 데이터 신호선: D0, D1, D2, ...
 - 주소 신호선: A0, A1, A2, ...

하드웨어 기본 용어

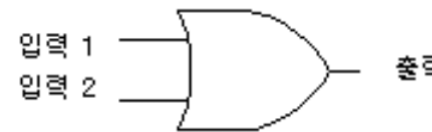
■ 논리 게이트

(1) AND 게이트



입력 1	입력 2	출력
High	High	High
High	Low	Low
Low	High	Low
Low	Low	Low

(2) OR 게이트



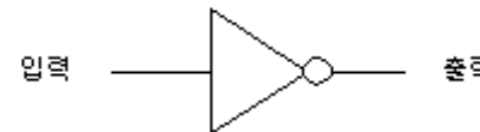
입력 1	입력 2	출력
High	High	High
High	Low	High
Low	High	High
Low	Low	Low

(3) XOR 게이트



입력 1	입력 2	출력
High	High	Low
High	Low	High
Low	High	High
Low	Low	Low

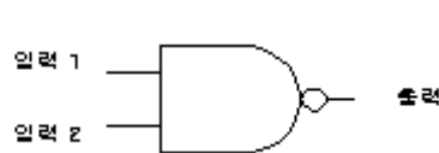
(4) 인버터



하드웨어 기본 용어

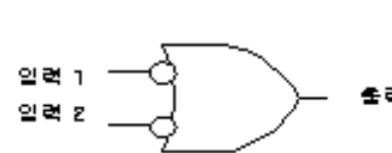
■ 논리 게이트

(5) NAND 게이트



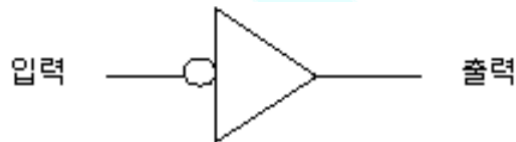
입력 1	입력 2	출력
High	High	Low
High	Low	High
Low	High	High
Low	Low	High

(6) 반대입력을 가진 OR 게이트

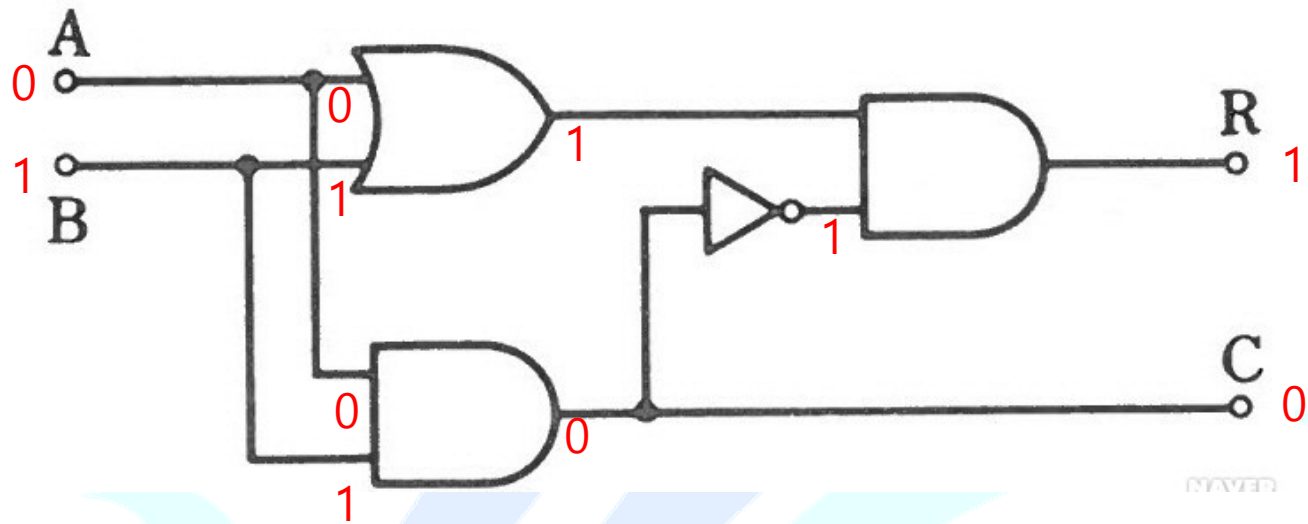


입력 1	입력 2	출력
High	High	Low
High	Low	High
Low	High	High
Low	Low	High

(7) 또 하나의 인버터



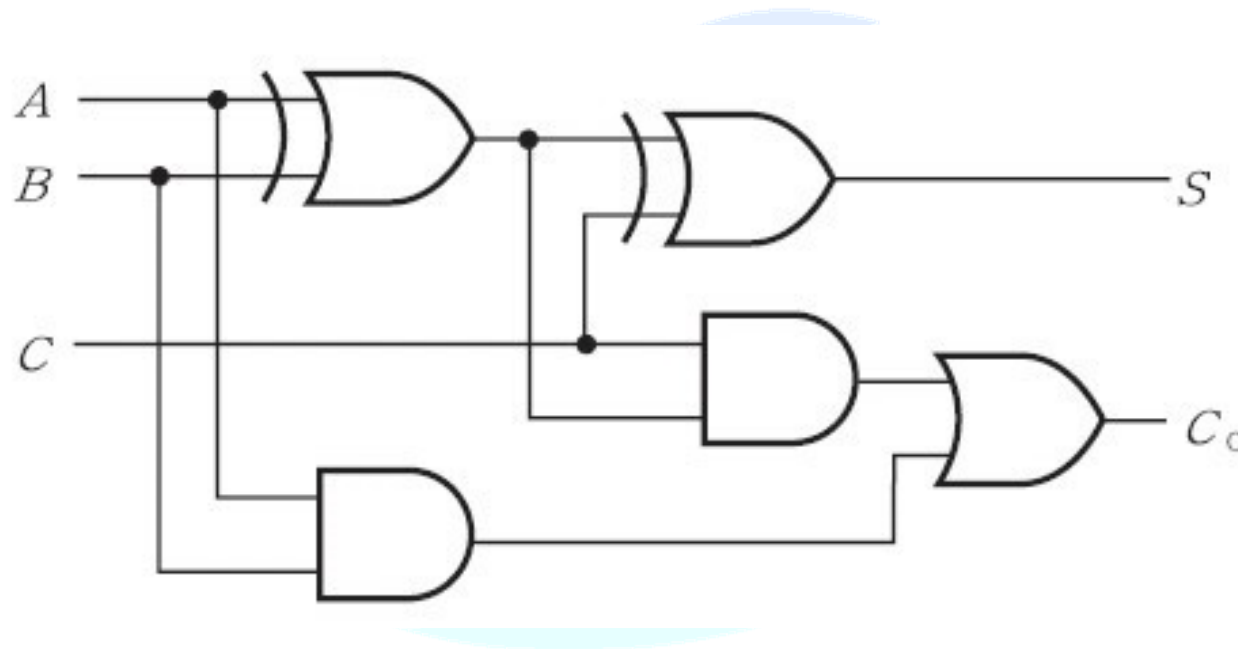
Example



입력		출력	
A	B	R	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

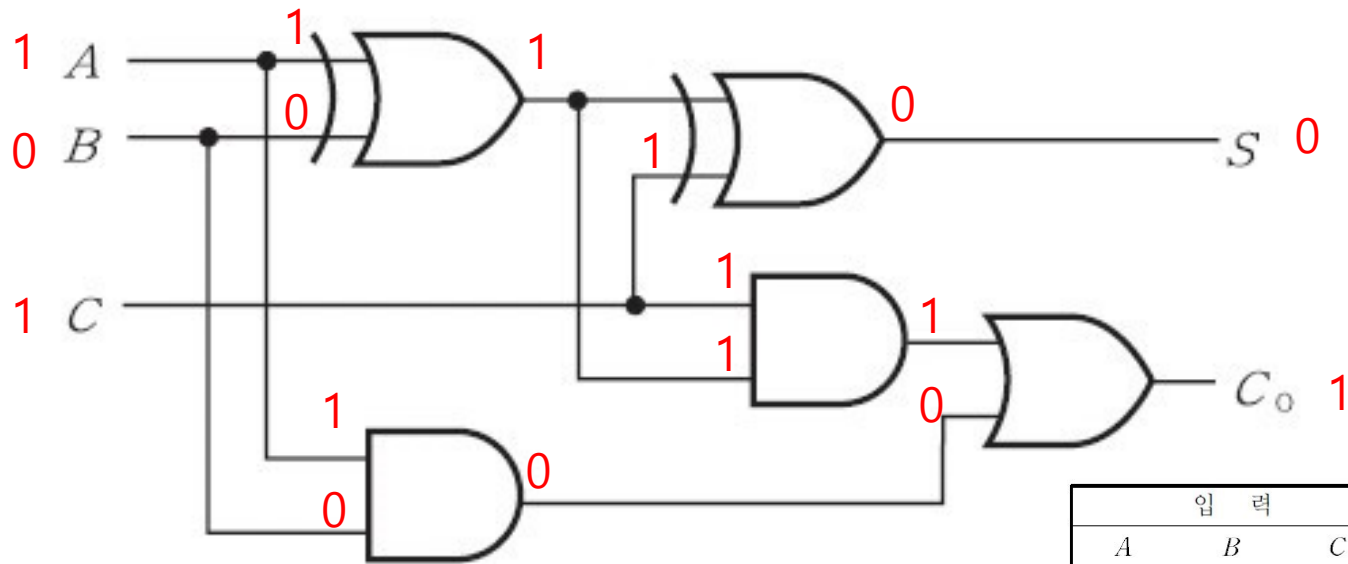
Practice

- 입력 $A=1$, $B=0$, $C=1$ 일 때 출력 S , C_0 의 값은?



Practice

- 입력 $A=1$, $B=0$, $C=1$ 일 때 출력 S , C_0 의 값은?

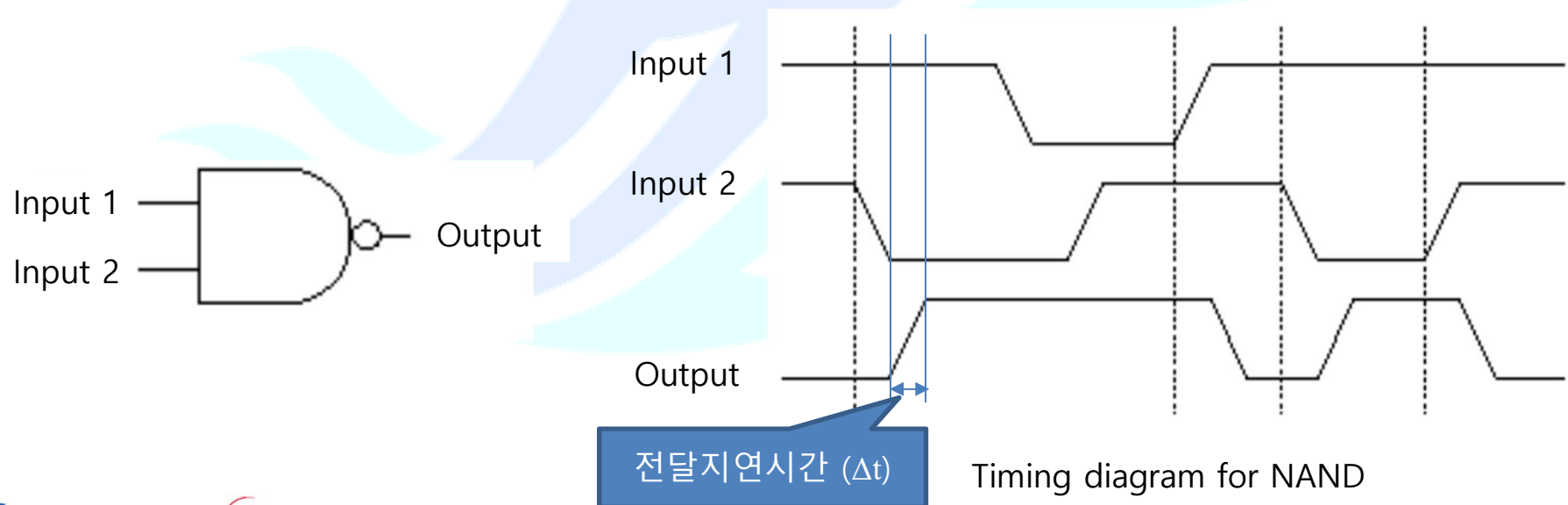


입 력			출 력	
A	B	C	C_0	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

하드웨어 기본 용어

■ 타이밍 다이어그램

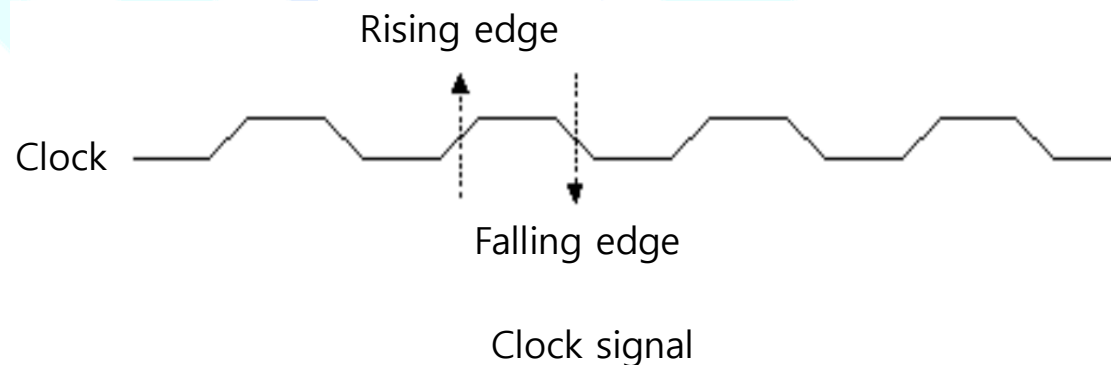
- 제조회사가 제조한 칩의 특성을 엔지니어에게 설명하기 위해서 사용하는 도구 중의 하나
- 각각의 입력과 출력의 변화와 서로간의 관계를 시간의 흐름 순으로 나타낸 그래프
 - 전달 지연시간: 입력이 바뀌면 출력이 언제 바뀌는지에 대한 시간 정보



하드웨어 기본 용어

■ 클럭 신호

- 회로의 장치들이 동작을 하기 위해서 Rising edge나 Falling edge를 공급해주는 역할
 - 회로가 어떠한 동작을 하기 위해서는 각 신호들의 상태가 변해야만 함
 - 어떤 임베디드 시스템 제품은 외부 이벤트가 입력을 변화시킬 때만 특정한 동작을 하기도 하지만, 대부분의 회로들은 시간의 흐름에 따라 동작을 수행
 - 마이크로프로세서로 구성되어 있는 회로는 외부 이벤트의 변화가 없어도 명령어들을 수행



하드웨어 기본 용어

- 오실레이터 & 크리스털
 - 클럭 신호를 발생시키는 장치
 - 오실레이터 : 스스로 클럭 신호를 발생시킴
 - 크리스털 : 두 개의 연결 핀을 가지고 있으며, 클럭 신호를 발생시키기 위해서 주변 회로가 필요함
 - ex.
 - 16MHz의 마이크로프로세서
 - 마이크로프로세서가 16MHz의 클럭에서 동작함
 - 더 빠른 클럭에서는 동작하지 않음

