6장. 메모리와 프로그램 논리소자

오늘의 교훈

불은 하늘로 오르려 하지만 결국 오르지 못하고, 물은 낮은곳으로 내려가지만
결국 하늘에 오른다.

메모리와 프로그램 논리소자의 정의

- 메모리
  - 2진값을 저장하는 다수의 셀(cell) 과 정보를 저장하고 검색하는 회로로 구성됨
  - 두종류의 메모리
    - RAM (Random Access Memory)
      - 읽고 쓰기 가능
      - 전원이 꺼지면 데이타가 사라짐
    - ROM (Read Only Memory)
      - 읽기만 가능
      - 전원이 꺼져도 데이타가 남아있음
- 프로그램 논리소자 (PLD (Programmable Logic Device))
  - 프로그램 과정을 통해서 내부 게이트나 선로의 연결을 변경할수 있는소자
  - ROM도 프로그램 논리소자임
  - 그외 PLA, PAL, CPLD 나 FPGA 같은 프로그램 논리소자가 있음

RAM

- Random Access 와 Sequential Access Memory
  - random access memory
    - 저장 위치에 상관없이 접근시간이 동일
  - sequential access memory
    - 저장 위치에따라 접근시간이 틀림
    - 예) 자기디스크, 자기 테이프등
- 저장 단위
  - 1 byte : 8 bit
  - 1 word : 16bit 혹은 32bit 로구성됨
  - 메모리의 기본단위는 1 바이트
- 메모리의 블록다이아그램 (*그림 6-2)
- 1024 × 16 크기의 메모리 저장내용 (*그림 6-3)

RAM (Cont'd)

- 쓰기와 읽기 연산
  - 메모리에 저장하기 (쓰기) 절차
    - 저장 위치를 가리키는 주소값을 어드레스선에 가함
    - 데이타선에 저장할 데이타값을 올림
    - 쓰기신호를 활성
  - 메모리에서 가져오기 (읽기) 절차
    - 가져올 위치를 가리키는 주소값을 어드레스선에 가함
    - 읽기신호를 활성
- 메모리칩의 제어입력

(CS)	(R/W')	메모리 연산
0	X	변화없음
1	0	선택된 워드 쓰기
1	1	선택된 워드에서 읽기

RAM (Cont'd)

- 타이밍 신호 파형
  - access time (t_a) : address 가해진 순간부터 데이타가 나타나기 까지의 최대소요시간
  - write cycle time (t_w) : address 가해진 순간부터 쓰기완료까지의 최대시간
  - 예) CPU: 50Mhz 로동작 (20ns), 메모리: 65ns (t_a) 및 75ns (t_w)
    * 적어도 4개의 클럭펄스가 필요 (타이밍 그림 (*그림 6-4))
- 메모리의 속성
  - SRAM (Static RAM)
    * 2진 정보를 latch 에 저장
    * 전원이 공급되는한 데이타 보존
    * 읽기쓰기 사이클이 짧음, 소용량
  - DRAM (Dynamic RAM)
    * capacitor 에 충전상태로 저장
    * 방전됨으로 주기적으로 refreshing 하여 데이타 보존
    * 읽기쓰기 사이클이 길음. 대용량
  - 비휘발성 메모리 : 자기적으로 저장하는 메모리 나 ROM
  - 휘발성 메모리 : SRAM, DRAM

RAM 집적회로

- SRAM cell의 내부구조
  - 1bit 를 저장하기 위한 회로도 (*그림 6-5)
  - SRAM bit slice (*그림 6-6)
  - 16 × 1 구조의 RAM chip (*그림 6-7)
  - 3-state buffer (*그림 6-8)
    - multiplexed output 이 가능하게 만듦 (*그림 6-9)
- 직교형 선택방식
  - $k \times 2^k$ decoder 는 2^k의 AND gate 가필요
  - 읽기/쓰기 공유 RAM cell 의 숫자가 매우커짐
    →access time 및 write time 이 지연됨
  - 해결방법 : 직교형 선택방식 (coincident selection) 사용
    - $k/2$ 개의 입력을 갖는 2개의 디코더를 사용
    - 두개의 디코더는 row select 와 column select 로 사용
- 직교형 선택방식의 예)
  - 16 × 1 RAM 의 block diagram (*그림 6-10)

RAM 집적회로 (Cont'd)

  - 8 × 2 RAM 의 block diagram (*그림 6-11)

- RAM chip 의 배열
  - 64K × 8 RAM의 기호 (*그림 6-12)
  - 256K × 8 RAM의 block diagram (*그림 6-13)
  - 64K × 16 RAM의 block diagram (*그림 6-14)

에러검출 및 교정

- 에러발생
  - RAM chip 내부의 트랜지스터나 커패시터는 아주작아 저장된 정보에서 에러발생
  - DRAM 에서 특히 많이 발생
- 해결방법
  - 에러검출기법과 에러교정부호 사용하여 해결
  - 가장 일반적인 방법: 패리티 비트방식
  - 데이타와 함께 저장되어 읽을때 확인
- 해밍코드
  - 가장 많이 쓰이는 에러교정코드
  - n 데이타 비트 + k 패리티비트 저장

에러검출 및 교정 (Cont'd)

– 예) 데이타가 8비트로 11000100 일때

비트위치	1	2	3	4	5	6	7	8	9	10	11	12
저장데이타	P_1	P_2	1	P_4	1	0	0	P_8	0	1	0	0

* 패리티 비트생성

  P_1 = (3,5,7,9,11)비트의 XOR = 0

  P_2 = (3,6,7,10,11)비트의 XOR = 0

  P_4 = (5,6,7,12)비트의 XOR = 1

  P_8 = (9,10,11,12)비트의 XOR = 1

* 저장 데이타

비트위치	1	2	3	4	5	6	7	8	9	10	11	12
저장데이타	0	0	1	1	1	0	0	1	0	1	0	0

에러검출 및 교정 (Cont'd)

* 패리티 체크
  C_1 = (1,3,5,7,9,11)비트의 XOR
  C_2 = (2,3,6,7,10,11)비트의 XOR
  C_4 = (4,5,6,7,12)비트의 XOR
  C_8 = (8,9,10,11,12)비트의 XOR
* 에러체크
  $C = C_8C_4C_2C_1 = 0000$ 이면 에러없음
* 에러 예)

비트위치	1	2	3	4	5	6	7	8	9	10	11	12	에러
저장데이타	0	0	1	1	1	0	0	1	0	1	0	0	없음
저장데이타	1	0	1	1	1	0	0	1	0	1	0	0	1번비트
저장데이타	0	0	1	1	0	0	0	1	0	1	0	0	5번비트

에러검출 및 교정 (Cont'd)

에러	C_8	C_4	C_2	C_1
없음	0	0	0	0
1번비트	0	0	0	1
5번비트	0	1	0	1

- n 과 k 의 관계식
  - $n + k \leq 2^k - 1$ 이어야함
  - $k = 3$ 이면 $n + k \leq 7$ 이고 $n \leq 7 - 3 = 4$ 이어야함
  - $k = 4$ 이면 $n + k \leq 15$ 이고 $n \leq 15 - 4 = 11$ 이어야함
- 해밍코드의 특성
  - 한비트에대한 에러만을 검출하고 교정할수 있다
  - 한비트의 패리티비트를 추가하여 하나의 에러교정, 두비트에러 검출할수 있다

에러검출 및 교정 (Cont'd)

* 예) 001110010100P_{13} 일때

C	P	상황
0	0	에러없음
$\neq 0$	1	1개에러, 교정가능
$\neq 0$	0	2개에러, 검출가능, 교정불가능
0	1	P_{13} 에서 에러발생

프로그램 논리소자

- 프로그램 논리소자들
  - ROM (Read Only Memory)
  - PLA (Programmable Logic Array)
  - PAL (Programmable Array Logic)
  - CPLD (Complex Programmable Logic Device)
  - FPGA (Field-Programmalbe Gate Array)
- 프로그램 방식
  - fuse 를 사용
    * 가장오래된 방법으로 초기에는 연결(closed) 상태
    * fuse 를 높은전압을 이용하여 끊음
  - mask programming
    * 제조과정에서 연결을 결정하여 연결
    * 비용비쌈, 동일한 프로그램을 대량으로 만들때 사용
  - antifuse 방식
    * 초기에는 끊긴(open) 상태

프로그램 논리소자 (Cont'd)

  * 고저항에 높은전압을 가하면 연결됨
- 위의 세가지 방식은 영구적방식 →한번프로그램하면 고치거나 지울수 없음
- SRAM bit 방식
  * SRAM에 연결정보를 저장
  * SRAM의 출력이 n channel MOS의 게이트를 조정
  * 휘발성 방식 →전원이 꺼지면 프로그램이 사라짐
- 트랜지스터의 스위치제어
  * floating gate 를 이용
  * floating gate 에 음전하를 충전 →open 상태
  * 높은전압으로 음전하를 제거 →closed 상태
    →electrically erasable
  * 프로그램지울때 자외선이나 높은전압을 사용

ROM

- 영구적으로 2진정보 저장, 비휘발성 소자
- 회로도 (그림 6-15) : k개의 주소선, n개의 데이타출력
- 32 $\times$ 8 ROM의 내부논리도 (*그림 6-16)
- ROM의 진리표 (*표6-2) 및 ROM의 프로그램 결과 (*그림 6-17)
- 4가지 프로그램 방식
  - fuse 방식의 ROM $\rightarrow$PROM $\rightarrow$한번 프로그램 가능
  - floating gate 방식의 ROM $\rightarrow$몇번이고 재프로그램 가능
    - EPROM $\rightarrow$자외선으로 지움
    - EEPROM $\rightarrow$전기적으로 지움
- 조합논리회로 구현
  - ROM : decoder + OR gate
  - 그러므로 ROM으로 조합논리회로 구현가능
  - 예제 6-1) 3비트의 2진수입력, 제곱을 2진수로 출력
    - ROM 진리표 (*표6-3)

ROM (Cont'd)

  * ROM 구현결과 (*그림 6-18)

- 세가지 주요 PLD
  - 특성

type	AND 항	OR 항
PROM	고정	프로그램
PAL	프로그램	고정
PLA	프로그램	프로그램

  - 구성도 (*그림 6-19)

PLA

- 3입력, 4곱항, 2출력 PLA (*그림 6-20)
  - 구현되 부울함수

    $F_1 = AB' + AC + A'BC'$

    $F_2 = (AC + BC)'$
  - PLA에 대한 프로그램표 (*표 6-4)
- 예제 6-2)
  - PLA로 다음두개의 부울함수를 구현하라

    $F_1(A, B, C) = \sum_m(0, 1, 2, 4)$

    $F_2(A, B, C) = \sum_m(0, 5, 6, 7)$
  - 풀이 (*그림 6-21)

PAL 소자들

- 대표적인 PAL의 내부 구성도 (*그림 6-22)
- 구현예)
  - 구현 부울함수

    $W(A,B,C,D) = \sum_m(2,12,13)$
    $X(A,B,C,D) = \sum_m(7,8,9,10,11,12,13,14,15)$
    $Y(A,B,C,D) = \sum_m(0,2,3,4,5,6,7,8,10,11,15)$
    $Z(A,B,C,D) = \sum_m(1,2,8,12,13)$
  - PAL 프로그램 표 (*표 6-5)
  - 구현된 PAL 내부연결도 (*그림 6-23)

VLSI 프로그램 소자들

- 필요성
  - PLD 는 복잡한 함수를 하나의 IC로 구현가능하게 함
  - 규모가 더 복잡한 함수를 구현할때 →VLSI 기술활용
  - VLSI 기술로는 수천에서 수백만까지의 게이트를 갖는 디지탈회로 설계가능
- VLSI 회로설계 방식
  - 완전 주문 맞춤 방식 (full custom design)
    - 고비용, 고집적, 고속, 대량의 IC를 제작 판매시 적합
  - 표준셀 설계방식 (standard cell design)
    - 중간비용, 중간성능, 기 설계된 회로를 구현
  - 게이트배열 (gate array) 방식
    - 실리콘내에 제작된 게이트들의 패턴
    - 1,000 에서 100,000개 정도의 배열이 하나의 IC로 제작됨

VLSI 프로그램 소자들 (Cont'd)

- VLSI 프로그램 논리소자
  - CPLD (complex programmable logic device)
  - FPGA (field-programmable gate array)
  - 특징
    - 많은 부분이 프로그램 가능한 조합논리로 되어있음
    - 미리 제작된 플립플롭을 내장
    - 조합회로와 플립플롭및 입출력에대해 프로그램가능한 연결부분을 갖음
    - 제작사마다 다른 특성을 갖음

VLSI 프로그램 소자들 (Cont'd)

- Altera MAX 7000 CPLDs
  - Altera 사가 제작
  - EEPROM의 floating gate 기술에 기초하여 제작
  - 전체구조 (*그림 6-24)
  - 특성
    * 16개의 논리배열 블럭이 있음
    * 각 논리배열의 출력이 프로그램 연결배열에 입력됨
    * 외부 입출력 블럭에서 입력을 받을수 있음
    * 프로그램배열 부분은 프로그램될수 있음
    * 각 논리배열부분은 16개의 매크로셀(macrocells)로 구성됨
    * 각각의 매크로셀은 1개의 플립플롭과 기본적인 조합논리로 구성
    * 매크로셀 내부의 AND게이트는 플립플롭의 제어를 위해서 사용
    * 플립플롭 자체도 프로그램될수 있음 (D,T,JK,SR형이 가능)
    * 다른 매크로셀의 AND게이트를 이용할수 있음
    * 모든 매크로셀의 입력에서 NAND게이트를 공통으로 사용가능

VLSI 프로그램 소자들 (Cont'd)

- Actel ACT 3 FPGAs
  - 게이트배열과 유사한 구조
  - 내부구조 (*그림 6-25)
  - 특성
    - antifuse 기술사용
    - 영구적이며 비휘발성의 프로그램을 형성
  - 두가지 논리모듈(*그림 6-26)
    - C module 과 S module 로 구성
    - 논리모듈내에서는 프로그램하지 못함
    - 연결선 트랙에 있는 antifuse 이용하여 프로그램

VLSI 프로그램 소자들 (Cont'd)

- Xilinx XC4000의 구조
  - XC4000 FPGA의 구조 (*그림 6-27)
  - CLB(configurable logic block) 라는 프로그램가능한 블록의 배열로 구성
  - switch matrices 를 이용하여 블록간 연결함
  - SRAM 을 사용하여 정보를 저장
  - 전원이 가해진후 내부 PROM으로 부터 자신에게 정보를 로드
    $\rightarrow$시간에 따라 다른기능을 하는 논리구현이 가능함
  - SRAM 비트가 논리값을 제어하는 세가지 방식 (*그림 6-28)
  - Xilinx 연결방식
    * CLB사이나 CLB와 IOB 사이의 연결은 수평및 수직채널 내의 선로 세그먼트를 통해서 이루어짐
    * long lines 라는 2개의 세그먼트가 있음
    * 스위치 매트릭스를 이용하여 연결됨
    * 스위치 매트릭스의 예 (*그림 6-29)
  - Xilinx 논리회로

VLSI 프로그램 소자들 (Cont'd)

* CLB와 IOB내에 존재
* CLB에 대한 개략도 (*그림 6-30)
* IOB 구조 (*그림 6-31)