

4장. 순서회로

오늘의 교훈

고난의 뿌리는 경쟁상의 성공을 행복의 주요한 원천이라고 지나치게 강조하는데서 돌아난다.

순서회로의 정의

- 순서회로란?
 - 기억소자나 feedback 이 있는 회로 (*그림 4-1)
 - 저장되는 정보는 해당시간의 상태(state)를 정의
 - 출력은 현상태 와 현재 입력값으로 결정됨
 - 기억소자 값은 전상태와 현재 입력값으로 결정됨
- 순서회로의 형태
 - 동기식 : 동기신호에 동기되어 동작
 - 비동기식 : 동기신호가 없다.

순서회로의 정의 (Cont'd)

- 저장장치
 - 버퍼 (*그림 4-2)
 - * t_{pd} 의 전파지연을 갖음 $\rightarrow t_{pd}$ 동안만 저장됨
 - 긴 시간의 정보저장 및 전파지연 상관없는 저장장치 필요
 - 저장방법 \rightarrow 페루프를 이용 (*그림 4-2 (b)(c)(d))
 - * 저장은 할수 있어도 저장된 데이터를 바꿀수는 없음
 - * 해결방법 \rightarrow NOR 나 NAND 게이트를 이용함 \rightarrow 래치 (비동기식 기억회로)
 - * 특정순간에만 데이터를 변화하게 하자 \rightarrow 클럭펄스 (clock pulses) 이용 (동기식 회로)
 - 클럭형 순서회로 (clocked sequential circuits)
 - * 클럭을 사용한 순서회로
 - ☛ 플립플롭 (flip-flop) 이라고 부름 (*그림 4-3)

래치 (latch)

- SR 과 S'R' 래치
 - NOR 게이트로 만들어진 래치 (*그림 4-4)
 - SR 래치 동작의 논리 시뮬레이션 (*그림 4-5)
 - NAND 게이트로 만들어진 S'R' 래치 (*그림 4-6)
 - 제어입력을 갖는 SR 래치 (*그림 4-7)
- D 래치
 - SR 래치에서 정의되지 않는 상태가 제거됨 (*그림 4-8)
 - 전송게이트를 갖는 D 래치 (*그림 4-9)

플립플롭

- 래치의 제어입력에 클럭펄스를 입력
 - 클럭이 1 인동안에는 입력변화가 출력을 변화시킴 →불안정한 동작
 - 해결책
 - * 두개의 래치를 연결 →마스타슬레이브 플립플롭
 - * 클럭이 전이되는 동안에만 구동 →edge-triggered 플립플롭
- 마스타슬레이브 플립플롭
 - 2개의 래치와 1개의 인버터로 구성됨 (*그림 4-10)
 - 논리 시뮬레이션 (*그림 4-11)
- JK 플립플롭
 - SR의 정의되지않는 문제 해결 (*그림 4-12)

플립플롭 (Cont'd)

- 에지트리거드 플립플롭
 - 클럭의 전이동안만 반응하는 플립플롭
 - 전이
 - * 클럭이 0에서 1로 변할때 동작 →positive edge triggered
 - * 클럭이 1에서 0로 변할때 동작 →negative edge triggered
 - D 형 positive edge triggered 플립플롭 (*그림 4-13)
 - 시간
 - * setup time : 클럭전이에 앞서 입력 D가 일정한값으로 유지되어야 하는 최소시간
 - * hold time : D 입력이 사용된후 변화되지 않아야 하는 최소시간
 - positive edge triggered JK 플립플롭 (*그림 4-14)
- 특성표 (*표 4-1)

플립플롭 (Cont'd)

- direct 입력
 - 클럭입력과 무관하게 비동기적으로 플립플롭을 세팅하거나 리셋팅 하기 위한 입력
 - * 1 로 세팅 하는 입력 : direct set 혹은 preset
 - * 0 로 세팅 하는 입력 : direct reset 혹은 clear
 - 전원이 켜질때등 정상적인 클럭에 반응하기전에 플립플롭을 초기화시 유용
 - direct set 과 reset 을 갖는 JK 플립플롭 (*그림 4-16)

순서회로 분석

- 입력식

- 플립플롭 입력식

- 예)

$$J_A = (XB + Y'C)$$

$$J_B = (YB' + C)$$

- 입력식에 대한 구현 (*그림 4-17)

- 순서회로의 예 (*그림 4-18)

- * 입력식

$$D_A = (AX + BX)$$

$$D_B = A'X$$

- * 출력식

$$Y = (A + B)X'$$

- 상태표 (*표 4-2, 4-3)

순서회로 분석 (Cont'd)

- 상태식

$$A(t+1) = D_A = AX + BX$$

$$B(t+1) = D_B = A'X$$

- 출력식

$$Y = AX' + BX'$$

● 모델

- 밀리 모델 (Mealy model) : 출력이 현재상태와 입력에 의존

- 무어 모델 (Moore model) : 출력이 현재상태에만 의존

- 무어 모델의 예)

* 입력식 : $D_A = A \oplus X \oplus Y$

* 출력식 : $Y = A$

* 논리도와 상태표 (*그림 4-19)

순서회로 분석 (Cont'd)

- JK 플립플롭을 이용한 분석
 - JK 플립플롭이 사용되었을때의 상태표 구하는 절차
 - * 현재상태와 입력의 향으로 각 플립플롭의 입력을 구함
 - * 특성식을 이용하여 다음상태 구함
 - 예)
 - * 입력식
$$J_A = B, K_A = BX'$$
$$J_B = X', K_B = AX' + A'X$$
 - * 상태표 (*표 4-4)
 - * 상태도 (* 그림 4-20)

순서회로 설계

- 설계절차

- 문제설명으로부터 상태도나 상태표를 얻음
- 상태에 2진 코드를 할당
- 플립플롭 입력식을 유도
- 출력식을 유도
- 입력식과 출력식을 논리 간략화
- 논리도를 그림

- 상태도와 상태표 작성법

- 예)

- * 문제설명

| |
|---|
| 1개의 입력과 1개의 출력을 갖고 입력이 110 이고 현재 입력이 1 일때 출력1을 발생, 그이외에는 출력 0 발생 |
|---|

- * 상태도를 얻음 (*그림 4-21)

- * 상태표를 얻음 (*표 4-5)(*표 4-6)

- 상태축소와 상태할당

순서회로 설계 (Cont'd)

- 상태축소 : 순차회로 설계시 상태수를 줄임 → flip-flop 이 줄어들 수 있음
- 상태할당 : 입출력 순서만 관심 → 임의의 상태 할당가능 → 조합회로 구성에 영향
- 상태축소
 - * 입출력 순서만 중요, 내부 상태는 임의여도 될 때
 - * 같은입력에 대하여 다음상태와 출력이 같은 것은 상태축소됨

순서회로 설계 (Cont'd)

| 현재상태 | 다음상태 | | 출력 | |
|------|-------|-------|-------|-------|
| | x = 0 | x = 1 | x = 0 | x = 1 |
| a | a | b | 0 | 0 |
| b | c | d | 0 | 0 |
| c | a | d | 0 | 0 |
| d | e | f | 0 | 1 |
| e | ☛a | ☛f | ☛0 | ☛1 |
| f | g | f | 0 | 1 |
| g | ☛a | ☛f | ☛0 | ☛1 |

* g와 e의 상태가 같음 : g 상태 제거후 g 상태를 e 상태로 대치

순서회로 설계 (Cont'd)

| 현재상태 | 다음상태 | | 출력 | |
|------|-------|-------|-------|-------|
| | x = 0 | x = 1 | x = 0 | x = 1 |
| a | a | b | 0 | 0 |
| b | c | d | 0 | 0 |
| c | a | d | 0 | 0 |
| d | ☛e | ☛f | ☛0 | ☛1 |
| e | a | f | 0 | 1 |
| f | ☛e | ☛f | ☛0 | ☛1 |

* f와 d의 상태가 같음 : f 상태 제거후 f 상태를 d 상태로 대치

순서회로 설계 (Cont'd)

* 결국 축소된 상태표

| 현재상태 | 다음상태 | | 출력 | |
|------|-------|-------|-------|-------|
| | x = 0 | x = 1 | x = 0 | x = 1 |
| a | a | b | 0 | 0 |
| b | c | d | 0 | 0 |
| c | a | d | 0 | 0 |
| d | e | d | 0 | 1 |
| e | a | d | 0 | 1 |

D 플립플롭을 이용한 설계

- 예제에 대한 상태도 (*그림 4-22)

- 예제에 대한 상태표 (*표 4-7)

- 입력식 및 출력식

$$A(t+1) = D_A(A, B, X) = \sum_m(2, 4, 5, 6)$$

$$B(t+1) = D_B(A, B, X) = \sum_m(1, 3, 5, 6)$$

$$Y(A, B, X) = \sum_m(1, 5)$$

- 논리간략화

$$D_A = AB' + BX'$$

$$D_B = A'X + B'X + ABX'$$

$$Y = B'X$$

- 논리도 (*그림 4-24)

● 사용되지 않는 상태를 갖는 설계

- 예제에 대한 상태표 (*표 4-8)

- 논리간략화 (그림 4-25)

JK 플립플롭을 이용한 설계

- 플립플롭 여기표 (excitation table) (*표 4-9)
- 설계절차
 - JK 플립플롭 입력을 갖는 상태표 (*표 4-10)
 - J와 K 입력식에 대한 맵 (그림 4-26)
 - 논리도 (*그림 4-27)
 - 시뮬레이션 결과 (*그림 4-28)
- Homework
 - 4장 연습문제
 - * 4, 5, 12, 13, 18, 19, 21, 27, 31