

PLD를 위한 순차회로 설계에 관한 연구

구 용 우*, 원 충 상**

충주 대학교 산업대학원 컴퓨터공학과

A study on sequential circuit design for PLD

Yong-Woo Koo*, Chung-Sang Won**

Dept of Computer Engineering, Graduate school chungju National University

E-mail: kooyw74@yahoo.co.kr, cswon@gukwon.chungju.ac.kr

요 약

순차 논리 회로를 설계하기 위하여 설계용 툴을 이용해 설계하고 그 결과를 시뮬레이션 한 것을 검토하여 실제 회로로 구현한다. 본 논문에서는 시뮬레이션을 하기 위한 회로설계 방법 중 Statatable를 작성하여 틀에 입력하고 심볼로 만들어, 이것을 이용하여 시스템을 구성 후 시뮬레이션을 하였다.

특히 설계 예제로 선택한 교통 신호제어 시스템은 대기차량이 없는 차로와 있는 차로를 구분하여 선택적으로 신호를 제공하므로써 교통소통의 효율을 물론 앞으로 교통 신호 제어체계의 새로운 모델을 제시하고 있다.

I. 序 論

일반적인 디지털 시스템은 그림의 흐름도와 같이 설계 계층 구조(Design Hierarchy)에 따라 시스템 레벨, 레지스터 레벨, 회로(circuits)레벨, 논리 레벨, 레이아웃(Layout) 레벨 등으로 나누어진다.

설계 단계가 상위 레벨에서 하위 레벨로 진행됨에 따라, 보다 구체적인 실제회로에 가까워지나 설계에 필요한 회로 정보와 필요한 시간은 더 많이 요구된다.

계층구조의 각 단계에서 회로를 설계할 때에는 각 단계에 적합한 기술 방법, 즉 하드웨어 설계 언어(HDL)

나 스키메틱 에디터(schematic editor) 등이 사용된다. 스키메틱 에디터를 사용한 설계 방법으로는 State diagram, Statetable, boolean equation, truthtable 등 여러 방법이 있으나 본 논문에서 Statetable을 사용하였다. 예제로 설계한 회로는 PLD를 사용하여 설계하기 위한 개선된 교통 신호제어 시스템을 설계하였다. PLD는 다양한 설계(CAD)툴과 하드웨어 에뮬레이터(Hardware Emulator)가 개발되어 있어 PLD를 이용하여 회로를 구현하는데 걸리는 시간을 크게 줄일 수 있게 되었다.

II. 상태도의 구현

1. 감지센서와 신호체계

상태표에 의한 디지털 회로 설계를 구현하기 위하여 네거리 교통 신호등 제어기에 대한 회로를 모델로 설계하였다. 일반적인 네거리의 교통 신호등은 정해진 순서와 시간간격으로 계속 반복 작동을 한다

이러한 신호체계는 교통량이 한쪽 방향으로 밀집되어 있는 시간 또는 장소에서는 원활한 교통 소통에 도움을 주지 못한다. 주행하는 차량이 없는 상황에서도 다른 차로의 차량은 정해진 신호가 진행 될 때까지 기다려야 하는 비율율이 발생한다.

교통여건을 파악하여 통행하는 차량이 없는 방향의 신호는 계속 통제하고 실제 운행하는 차량이 있는 방향의 신호를 계속 주어 주행하도록 한다면 소통효율은 높아 질 것이다.

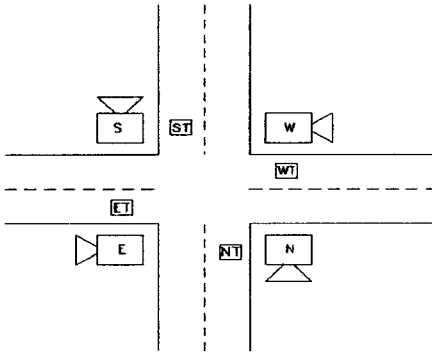


Fig. 1-1 차량 감지센서가 설치된 교차로의 시스템 개념

이러한 신호체계를 위하여 Fig 1-1 과 같은 차량 감지센서를 부착하여 대기 차량이 없는 방향의 통행 신호는 건너뛰고 대기 차량이 있는 방향의 차량은 주행하도록 교통신호의 변화를 주는데 그때의 감지신호 체크 우선 순위는 반 시계 방향(기본 신호기의 상태 변경순서와 동일)으로 하도록 하였다.

차량이 북쪽으로 주행을 하고 있다가 동쪽 방향 차로에만 차량이 대기하고 있는 경우 우선 순위에 의해서 서쪽과 남쪽의 차량감지 센서를 체크해 보고 없을 경우 서쪽과 남쪽의 신호는 계속 적색으로 하고 동쪽 방향에만 통행을 허가하는 청색 등을 점등한다.

이와 같은 방식의 신호제어는 각 방향의 차량감지 센서의 조건에 따라 어느 쪽 방향의 차량통행을 차단시킬 것인가, 또는 차량통행을 계속 유지시킬 것인가를 결정한다.

Fig 1-2는 차량 감지센서가 설치된 교차로의 교통 신호기의 상태도이다.

황색신호 상태인 차로에서 다른 쪽 방향의 차량 감지 센서가 대기 차량을 감지하지 못하면, 다시 청색등을 점등하므로 해서 정해진 한 주기의 신호를 더 주어지게 된다.

황색 신호를 기준으로 청색 신호의 비율을 1 : 16으로 하였고 청색, 황색 이외의 모든 신호는 적색이 점등 되도록 하였다. 그러나 이들 중 어느 하나의 센서가 차량을 감지하게 되면 Fig. 1-2의 상태로 동작하는 제어기에 신호가 전달되어 상태 변환의 신호로 이용된다. 이러한 센서의 신호를 입력신호(Input Signal)라고 하였다.

동작순서의 결정에 필요한 입력신호와 결정된 동작을 수행시키는 출력신호, 동작의 시간을 지정해 주는 클럭 신호(Clock Signal)를 기본적인 신호로 사용한다.

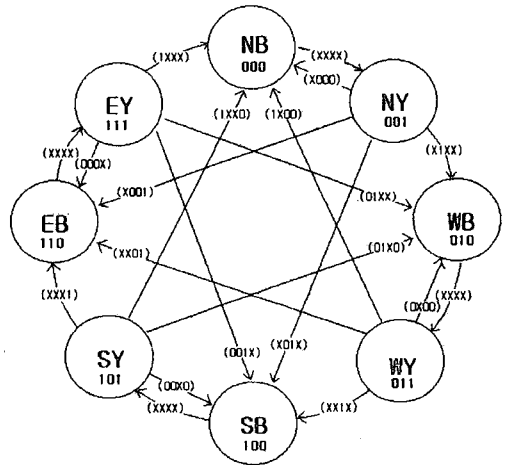


Fig 1-2 차량 감지 센서 신호를 고려한 상태도

상태도에 나타낸바와 같이 항상 황색 신호가 발생하는 시점에서 각 방향의 센서 입력을 감지하여 다음 상태가 결정 되도록 하였다.

WY(011)상태에서 센서 입력 0X00이면 다시 WB

(010)상태가 되고 센서 입력이 XX1X이면 SB(100) 상태가 되며 센서 입력이 1X00이면 NB(000)상태, 센서입력이 XX01이면 EB(110)상태로 다음 상태가 결정 되도록 하여 대기 차량이 없는 차로에 불필요한 주행 신호를 생략하였다.

현재상태			센서 입력			다음상태			신호기 출력											
A	B	C	N	S	E	A'	B'	C'	NB	NY	NR	NB	NY	NR	SB	SY	SR	EB	EY	ER
0	0	0	X	X	X	0	0	1	1	0	0	0	0	1	0	0	1	0	0	1
0	0	1	X	1	X	0	1	0	0	1	0	0	0	1	0	0	1	0	0	1
0	0	1	X	0	1	X	1	0	0	0	1	0	0	0	1	0	0	1	0	0
0	0	1	X	0	0	1	1	1	0	0	1	0	0	0	1	0	0	1	0	0
0	0	1	X	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0
0	1	0	X	X	X	0	1	1	0	0	0	1	1	0	0	0	1	0	0	1
0	1	1	X	X	1	X	1	0	0	0	0	1	0	1	0	0	0	1	0	0
0	1	1	X	X	0	1	1	1	0	0	0	1	0	0	1	0	0	0	1	0
0	1	1	1	X	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0
0	1	1	0	X	0	0	0	1	0	0	0	1	0	1	0	0	0	1	0	0
1	0	0	X	X	X	1	0	1	0	0	0	1	0	0	1	1	0	0	0	0
1	0	1	X	X	X	1	1	1	0	0	0	1	0	0	1	0	1	0	0	0
1	0	1	1	X	X	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0
1	0	1	0	1	X	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0
1	0	1	0	0	X	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0
1	1	0	X	X	X	1	1	1	0	0	0	1	0	0	1	0	0	1	1	0
1	1	1	1	X	X	0	0	0	0	0	0	1	0	0	1	0	0	1	0	1
1	1	1	0	1	X	X	0	1	0	0	0	1	0	0	1	0	0	1	0	1
1	1	1	0	0	1	X	1	0	0	0	0	1	0	0	1	0	0	1	0	1
1	1	1	0	0	0	X	1	1	0	0	0	1	0	0	1	0	0	1	0	1

표 1 교통제어기 출력신호의 상태표

2. 상태표의 입력

Fig. 1-2 와 같은 상태를 직접 틀에 입력시켜 설계하는 방법도 있으나 본 논문에서는 Mylogic station을 사용하구 상태표를 사용한 설계를 위하여, 표 1과 같은 상태표를 추출하였다.

MyLogic Station package 프로그램 메뉴에서 Schematic Editor(SchEd) 아이콘을 선택하여 프로그램을 실행한다. Logic Simulator, EDIF Generator 및 Schematic Generator는 Schematic Editor의 도구 메뉴에 연결되어 있어 Schematic Editor에서 바로 실행이 가능하다.

MyLogic Station tool에서 생성되는 MDC File을 윈도우 탐색기나 검색 창에서 직접 더블 클릭하여도 실행이 된다.

새 디자인 만들기 와 라이브러리 추가 등 필요한 과정을 거쳐 셀 만들기에서 셀의 이름을 지정한 다음 뷰 만들기에서 Fig. 2 -1 과 같이 여러 종류의 뷰 중에 state table을 선택한다

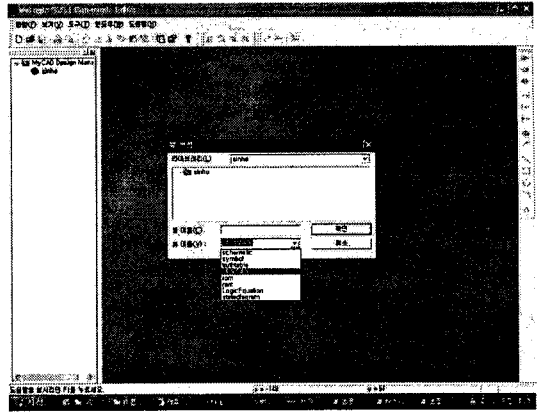


Fig. 2-1 Schematic Editor에서의 뷰 만들기

사용자가 디자인할 뷰가 생성되고 틀은 상태표 입력을 위한 정보를 선택하는 화면이 나타나게 된다.

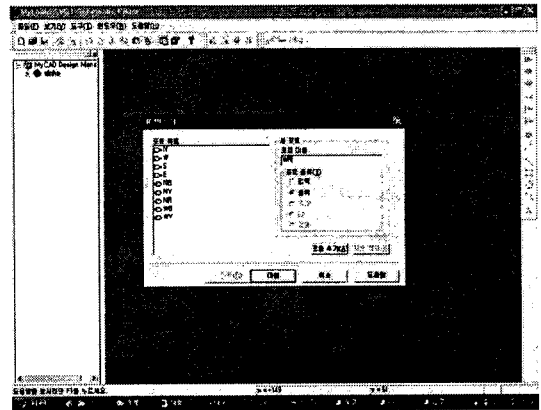


Fig. 2-2 Schematic Editor에서 포트등록

입력포트 설정하기에서 포트이름, 포트종류가 정해지고 포트추가를 통해 포트목록에 등록이 된다.

이렇게 포트 등록이 끝나면 표1 교통신호 제어기 상태표를 입력하게 된다.

첫 번째 Data는 각 방향 센서의 입력과 회로의 동작 timing(rising edge 또는 falling edge)을 지정하는 부분이고, 두 번째 State는 현재 상태를 지정하는 부분이며 세 번째 Next State는 다음상태 값을 지정하며, 네 번째 Result는 결과(출력)값을 지정하는 부분이다. 상태표에 각 상태 값만 입력시켜 주면 별도의 Design 없이 바로 Symbol화하여 회로 설계에 사용할 수 있다.

Steps	NW SE CLK	A B C	XYZ	Next State	Result
1	---W	000	001	100001001001	
2	-1-W	001	010	010001001001	
3	-01-W	001	100	010001001001	
4	-001W	001	110	010001001001	
5	-000W	001	000	001010001001	
6	---W	010	011	00100001001	
7	--1-W	011	100	001010001001	
8	-01W	011	110	001010001001	
9	1-00W	011	000	001010001001	
10	0-00W	011	010	001010001001	
11	---W	100	101	001001100001	
12	--1W	101	110	001001010001	
13	1-0W	101	000	001001010001	
14	01-0W	101	010	001001010001	
15	00-0W	101	100	001001010001	
16	---W	110	111	001001001100	
17	1--W	111	000	001001001100	
18	01--W	111	010	001001001100	
19	00--W	111	100	001001001100	
20	000-W	111	110	001001001100	

Fig. 2-3 Schematic Editor에서 상태표를 입력한 화면

이러한 상태표의 입력을 통해 Symbol 생성을 하여 추후 설계에 활용할 수 있으며 Fig. 2-4는 이러한 과정을 통해 생성된 상태표에 의한 Symbol을 보여주고 있다.

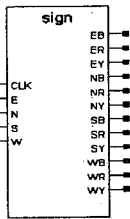


Fig. 2-4 상태표로 생성된 상태제어 심볼

3. 가변 클럭 제어회로

신호동의 시간적 신호 체계를 규정하기 위하여 일단 청색 신호가 주어지면 앞에서 설명한 황색 신호의 정수 배가 되도록 하였다.

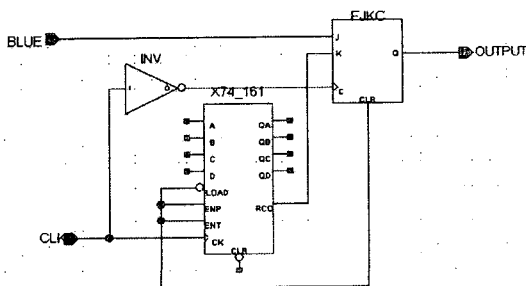


Fig. 3-1 가변 클럭 발생회로

일단 출력 신호 중 어느 방향이든 청색신호가 검출되면 16 클럭 사이클이 지속되어야 하므로 필요한 회로 설계를 위해 병렬로드를 가진 4bit 카운터를 사용하여 0에서 15까지의 카운팅을 하였고 이후 발생하는 carry신호를 JK-FF의 K 입력에 연결하여(즉 reset) 필요한 신호를 구하도록 하였다.

4. 생성된 회로의 동작

상태표의 입력에 의해 만든 심볼과 가변 클럭 발생회로의 조합에 의해 생성된 교통 신호 제어 시스템을 설계하였다.

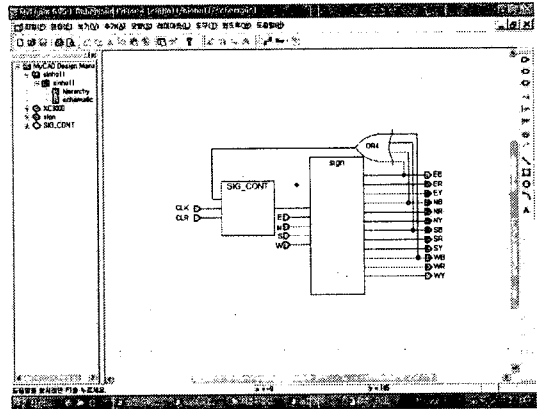


Fig. 4-1 생성된 심볼로 설계한 시스템 회로

본 논문에서 중점을 둔 각 방향의 센서 입력을 Fig 4-2와 같이 다양하게 주어 네 방향 모두 대기 차량이 있는 일반적인 경우와 한 방향, 두 방향, 세 방향, 네 방향 모두 대기 차량이 없는 경우까지 시뮬레이션을 해보았다.

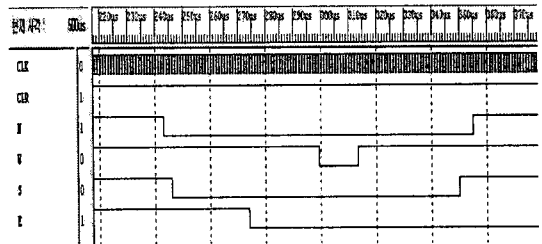


Fig 4-2 시뮬레이션시 입력 신호

그 결과 Fig 4-3과 같은 기대한 결과가 출력이 되었으며 다음 상태 결정 시 황색 등 점등 시간인 1 클럭

사이클 동안에 정확하게 다음상태를 결정하여 출력이 발생하는 만족한 결과를 얻을 수 있었다.

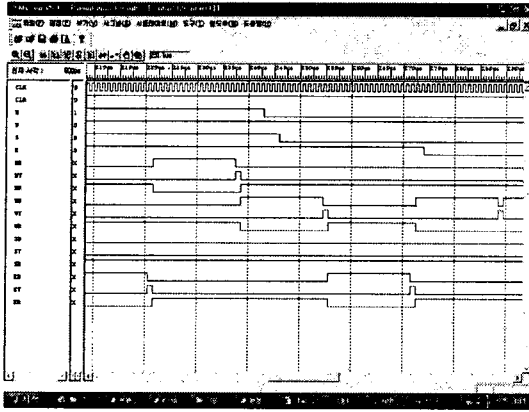


Fig. 4-3 시뮬레이션 결과

시뮬레이션과 회로에는 표시하지 않았으나 좌회전 청색신호는 직진 청색 신호와 같은 체계이므로 별도의 출력으로 지정하지 않았고 설치 시공시 두 신호를 같이 취급하면 각 방향 네 가지 신호가 표현 될 것이다

III. 結論

순차회로의 설계에서 비교 및 판단(Decision)시 문제가 되는 시간지연 문제를 상태표로 만든 심볼을 사용한 결과, 1 사이클만에 다음상태가 결정될 수 있었으며 비교적 간단한 가변 클럭 제어회로로 필요한 신호 Timing을 제어 할 수 있었다.

본 논문에서는 PLD, 특히 FPGA등으로 설계할 수 있는 과정을 미리 시뮬레이션을 통해 검증하였으며 이 결과를 이용하여 즉시 실제 회로로 만들 수 있는 여건을 만들었다.

예제로 설계한 제어시스템은 차량 감지센서의 입력으로 교통 신호를 제어하므로써 좀더 효율적인 교통의 흐름을 제공할 수 있다는 것을 알게 되어 효율적인 신호등 제어기의 새로운 패러다임을 제시했다고 할 수 있으며 차후 연구과제로 이 논문에서는 차량의 대기 유무에 따라 신호를 주었지만, 차후에는 대기 차량의 수를 판독하여 대기 차량이 많은 경우 확일적으로 주던 신호의 Timing을 능동적으로 가변시켜 줄 수 있도록 하는 인공지능형 교통 신호 제어 시스템에 대해 연구가 필요하리라 생각된다.

參考文獻

1. "小林芳直" PLD 의 논리회로 설계법 p27~p34
2. 원 충상, " PLD 설계 System 개발에 관한 연구" 박사학위 논문집
3. 이 근만, 정 원섭 "디지털 시스템 설계" 대응
4. PARAG K,LARA, "PLD DIGITAL SYSTEM DESIGN USING PROGRAMMABLE LOGIC DEVICE" , Prentice-Hall, Inc. 1990
5. 김 희석, 원 충상, "PLD 설계용 틀 개발에 관한 연구",한국정보처리이용학회 논문집, 제1권 제3호 pp391-397, 1994
6. 김재진,김성무,조남경,변상준,원충상,김희석 "PLD partition을 고려한 PLD설계용 틀 개발" 대한전자공학회 추계종합학술대회 논문집, pp1415-1418, 1994
7. 원 충상 "PLD를 이용한 교통제어 시스템 설계" 충주 대학교 논문집 제34집 2호 1999, 2
8. 원 충상 "GAL을 이용한 디지털 시스템 설계" 충주대학교 논문집 36집 2호 2001. 11
9. 구용우, 원충상, 박재문, "GAL을 이용한 신호등 제어기 설계" 한국 멀티미디어 학회 추계학술발표논문집, pp827~832, 2001, 10