

## GAL을 이용한 신호등 제어기 설계

구 용 우\*, 박 재 문\*, 원 충 상\*\*  
충주 대학교 산업대학원 컴퓨터공학과

### A Design of the Traffic Light Controller using by GAL Devices

Yong-Woo Koo\*, Jae-Moon Park\*, Chung-Sang Won\*\*

\*Dept of Computer Engineering, Graduate  
school chungju National University

E-mail: kooyw74@yahoo.co.kr, jmpark@icols.com,

\*\*Dept of Computer Engineering chungju National University  
cswon@gukwon.chungju.ac.kr

#### 요 약

기존 신호등 체계의 문제점을 보완하기 위해 센서를 이용하여 각 차로에 진행하려고 하는 차량을 감지하고, 대기 차량이 없을 경우 다음방향으로 신호를 주어 소통에 효율성을 높였으며 특히 GAL소자 하나로 설계함으로써 개발비용을 줄이도록 하였다.

#### I. 序 論

반도체 소자의 성능과 집적도의 향상에 따라 필요한 회로를 단일 칩으로 설계하고 복잡한 설계과정을 컴퓨터를 이용한 툴을 사용함으로써 개발 기간의 단축은 물론 시스템 규모를 적게 개발할 수 있어서 많은 활용을 하고 있다.

개발된 회로의 대량 생산을 위하여 ASIC (Application Specific Integrated Circuit)으로 구현하므로 비용이나 규모를 최소화 할 수도 있고 또 비교적 큰 규모의 회로 설계는

FPGA(Field Programmable Gate Array)를 이용한 설계가 효과적이다.

그러나 개발 단계의 비교적 소규모의 회로나 소량의 수요에 특히 PLD(programmable Logic Device)를 이용한 설계가 훨씬 유리하다. PLD중에서도 PAL 계열은 한 번 프로그램하면 재사용을 할 수 없으나 GAL (Generic Array Logic) 계열의 소자는 반복 사용이 가능하여 경제적이며, 특히 22V10은 출력의 수를 10개까지 사용할 수 있어서 디지털 시스템을 설계하는데 유용하게 사용할 수 있다.

GAL을 이용한 설계는 PC를 기반으로 한 설계 툴을 사용하여 설계하므로 필요한 회로를 수요자가 직접 자신의 컴퓨터로 설계하여 GAL회로 구현에 필요한 target file인 JED

-EC(Joint Electron Device Engineering Council) file을 생성한다.

PC에 연결된 간단한 PLD 프로그램장치(AL L-07)를 통해 프로그램 한 것을 즉시 검증하여 회로구현의 결과를 확인하여 사용할 수 있고 필요한 경우 즉시 수정을 할 수 있다.

이러한 일련의 설계과정을 이용함으로써 회로의 신뢰도를 향상시키는 물론 저렴한 비용으로 시스템을 구현하여 개발 기간의 단축으로 인한 개발 원가를 절감할 수 있다.

GAL을 이용해 다양한 디지털 회로의 구현이 가능하다. 본 논문에서는 교차로인 네거리의 소통효율을 높이기 위해 각 차로마다 차량감지 센서를 설치하여 대기차량이 있는 차로와 대기차량이 없는 차로를 구분하여 통행할 차량이 없는 차로에 통행신호를 불필요하게 주어 차량소통에 지장을 주는 불합리한 신호 체계를 개선하고자 개선된 교통신호 제어기를 설계하였다.

교통신호 제어기를 현재상태와 다음상태를 고려한 상태표를 만들고 상태표에서 부울식을 유도하여 GAL소자 22V10을 사용하여 설계하였다.

## II. 제어기의 구현

### 1. 감지센서와 신호체계

일반적인 네거리의 교통신호등은 정해진 순서와 시간간격으로 계속 반복하여 작동을 한다. 다만 본 논문에서는 남북방향, 동서방향의 같은 신호체계를 가진 두 가지 패턴의 신호체계를 모델로 삼았다.

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 비효율이 발생한다.

교통여건을 파악하여 통행하는 차량이 없는 방향의 신호는 계속 통제하고 실제 운행하는

차량이 있는 방향의 신호를 계속 주어 통행토록 한다면 소통효율은 높아 질 것이다.

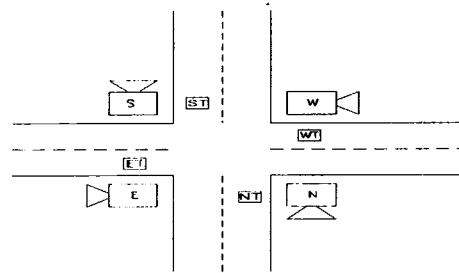


Fig 1-1 차량 감지센서가 설치된 신호기 개념

이러한 신호체계를 위하여 Fig 1-1 과 같은 차량 감지센서를 부착하여 동서방향으로 주행하는 차량이 감지되기 전까지는 남북방향의 신호등을 녹색으로, 동서방향의 신호등을 적색으로 고정시킨 다음, 동서방향 차로에 차량이 대기하고 있는 경우에만 남북방향의 신호를 차단시키는 제어를 한다. 다시 남북방향으로 주행하려는 차량이 감지되면 동서방향의 신호는 일정기간 녹색신호가 유지된 후 적색으로 점등되고 남북방향은 녹색으로 점등된다.

이와 같은 방식의 신호제어는 양쪽 차량감지센서의 조건에 따라 어느 쪽 방향의 차량통행을 차단시킬 것인가, 또는 차량통행을 계속 유지시킬 것인가를 결정하여야 한다.

Fig 1-2는 차량감지센서가 설치된 교차로의 교통신호기의 순서도이다.

적색신호 차로에서 차량감지센서가 차량을 감지하지 못하면, 주행 중인 방향의 녹색신호는 상태가 계속 유지된다. 그러나 이들 중 어느 하나의 센서가 차량을 감지하게 되면 Fig 1-2의 순서도로 동작하는 제어기에 신호가 전달되어 신호변환의 신호로 이용된다.

이러한 센서의 신호를 입력신호(Input Signal)라고 하였다.

동작 순서의 결정에 필요한 입력신호와 결

정된 동작 블럭의 동작을 수행시키는 출력 신호 및 동작순서의 시간을 지정해 주는 클럭 신호(Clock Signal)를 기본적인 신호로 사용한다.

Fig 1-2에 나타낸 것과 같이 각 상태는 5초간 지속되는 것을 하나의 동작 블럭으로 규정하고 입력신호가 감지되어도 즉시 소동 신호가 작동하지 않고 하나의 동작 블럭이 끝난 다음 입력신호 여부에 따라 다음의 동작 블럭으로 진행한다.

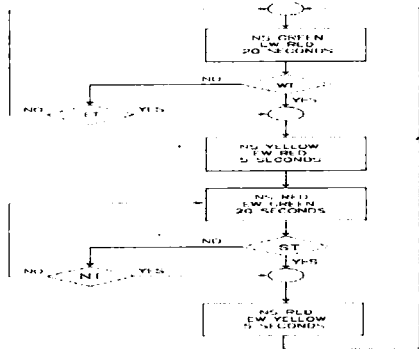


Fig 1-2 차량감지센서를 사용한 신호기의 순서도

## 2. 신호표기법

교통신호가 시간에 따라 변하는 모양을 나타내는 방법으로 Fig 2-1과 같은 타이밍 차트(timing chart)을 이용한다. 그림에는 교통제어기에 사용되는 6개의 신호가 표시되어 있다. 각 신호는 논리적으로 "1"과 "0"의 2가지 신호로, "1"은 그 출력이 점등(on)되어 있는 상태를 의미하며, "0"인 경우, 소등(off) 상태를 의미한다.

예를 들어 시간이 25초 경과된 시점에서 NSR 신호는 "off"에서 "on"상태로, NSY는 "on" 상태에서 "off"상태로 NSG는 "off" 상태를 계속 유지한다.

EWG 신호는 "on" 상태에서 "off"상태로 EWY 신호는 "off"에서 "on"상태로 바뀌고 EWR 신호는 "off" 상태를 그대로 유지한다.

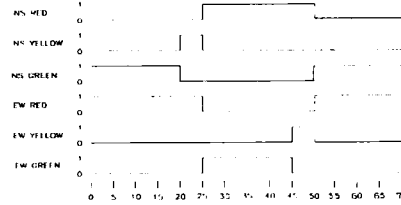


Fig 2-1 교통 제어기 출력 신호의 타이밍 차트

그러나 신호의 종류가 많은 경우 타이밍 차트로 표현하는 것 보다 신호를 도표의 형태로 나타내는 것이 보다 간결하다. 대부분의 디지털회로에서는 신호변화가 시간간격(clock interval)이라 불리는 일정한 시간대에서만 일어나는 특징을 갖고 있다.

Fig 2-1의 교통제어기 출력신호의 타이밍도가 이러한 특징을 갖고 있다. 신호의 변화는 5초 간격으로 일어나며 타이밍 차트로 표시되는 모든 출력신호는 매 5초 간격으로 신호상태를 도표형식으로 나열시킬 수가 있다. Table 1은 도표형식으로 기술한 상태표이다.

Table 1 교통제어기 출력신호의 상태표

	NSR	NSY	NSG	EWR	EWY	EWG
0	0	0	1	1	0	0
5	0	0	1	1	0	0
10	0	0	1	1	0	0
15	0	0	1	1	0	0
20	0	1	0	1	0	0
25	1	0	0	0	0	1
30	1	0	0	0	0	1
35	1	0	0	0	0	1
40	1	0	0	0	0	1
45	1	0	0	0	1	0
50	0	0	1	1	0	0
55	0	0	1	1	0	0
60	0	0	1	1	0	0
65	0	0	1	1	0	0
70	0	1	0	1	0	0

Table 2 출력신호의 논리적 의미

부호	신호의 논리가 "0"일때	신호의 논리가 "1"일때
NSG	남북방향 녹색등 소등	남북방향 녹색등 점등
NSY	남북방향 황색등 소등	남북방향 황색등 점등
NSR	남북방향 적색등 소등	남북방향 적색등 점등
EWG	동서방향 녹색등 소등	동서방향 녹색등 점등
EWY	동서방향 황색등 소등	동서방향 황색등 점등
EWR	동서방향 적색등 소등	동서방향 적색등 점등

### 3. 신호제어기의 구현

차량감지센서가 설치된 교통신호제어기의 구현을 위해 Fig 1-2의 순서도를 기반으로 각 동작 블록의 시간간격을 동일하게 설정하여 Fig 3-1의 순서도를 구한 후 순서도의 각 동작 블록에 상태코드를 할당시킨다.

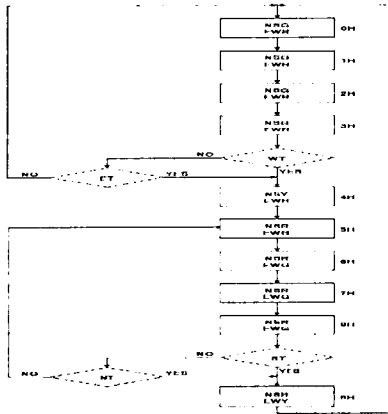


Fig 3-1 교통 제어기의 동작 블록 상태할당

상태 "3H"의 다음 상태는 센서 입력신호 ET와 WT의 조건에 따라 상태 "4H" 또는 상태 "0H" 가 될 수 있고, 상태 "8H"의 다음 상태는 센서 ST와 NT의 조건에 따라 상태 "5H" 또는 상태 "9H"가 될 수 있기 때문이다.

이러한 상황을 고려한 신호제어기를 GAL을 이용하여 설계를 할 수 있다. Fig 3-2와 같이 4개의 센서 입력과 clock pulse를 입력으로 사용하여 신호등의 6개의 출력(P0,P1,P2,P3,P4,P5)을 GAL소자의 출력으로 지정할 수 있다.

우선 Fig 3-1의 순서도로부터 Table 3의 상태표를 작성한다.

상태표에서 현재상태를 A, B, C, D로 표시하였고 다음 clock pulse와 입력신호 (ET, WT, ST, NT)에 의해 다음상태 A', B', C', D' 가 결정이 되며, 출력도 결정이 된다.

다만 설계 시 중요한 것은 현재상태와 다음상태는 GAL소자의 내부 Flip-Flop의 상태이므로 부울식 표현에는 당연히 표시되어야 하나 입출력 변수로는 지정할 필요가 없다. 즉

Fig 3-2의 블록 다이어그램에서처럼 4 입력과 6 출력으로 가능하다.

현재상태와 다음상태는 Mealy Model의 디지털 시스템으로서 현재상태가 feedback되어 외부 입력과 함께 다음상태와 출력을 결정하는 구조로 되어 있다.

설계에 사용할 GAL22V10의 블록 다이어그램을 Fig 3-3에 나타내었다.

GAL22V10의 특징은 그림에서 보는 바와 같이 출력 측에 지정할 수 있는 변수들이 PAL소자들처럼 획일적인 구조로 되어 있지 않고 만들 수 있는 product term의 수를 가변적으로 설치하여 8, 10, 12, 14, 16개 등 상황에 따라 적당한 출력을 선택하여 사용할 수 있는 특징과 출력 부분에 OLMC(Output Logic Macro Cell)를 통한 출력의 다양한 변화를 가능하게 할 수 있어 설계에 유연성과 실용성을 갖춘 소자라 할 수 있다.

Table 3 압축된 상태표

현재 상태	입력				다음 상태				NS	NS	NSR	EWL	EW1	EW2	
A B C D	E	W	S	N	A'	B'	C'	D'	P0	P1	P2	P3	P4	P5	
0 0 0 0	X	X	X	X	0	0	0	1	1	0	0	0	0	0	1
0 0 0 1	X	X	X	X	0	0	1	0	1	0	0	0	0	0	1
0 0 1 0	X	X	X	X	0	0	1	1	1	0	0	0	0	0	1
0 0 1 1	0	X	X	X	0	0	0	0	1	0	0	0	0	0	1
0 0 1 1	1	1	X	X	0	1	0	0	0	1	0	0	0	0	1
0 0 1 1	1	0	X	X	0	1	0	0	0	1	0	0	0	0	1
0 0 1 1	1	1	0	X	X	0	1	0	0	0	1	0	0	0	1
0 1 0 0	X	X	X	X	0	1	0	1	0	0	1	1	1	0	0
0 1 0 1	X	X	X	X	0	1	1	0	0	0	1	1	0	0	0
0 1 1 0	X	X	X	X	0	1	1	1	0	0	1	1	1	0	0
0 1 1 1	X	X	X	X	1	0	0	0	0	0	1	1	1	0	0
1 0 0 0	X	X	0	0	0	1	0	1	0	0	1	1	1	0	0
1 0 0 0	X	X	0	1	1	0	0	1	0	0	1	0	1	0	0
1 0 0 0	X	X	1	0	1	0	0	1	0	0	1	0	1	0	0
1 0 0 0	X	X	1	1	1	0	0	1	0	0	1	0	1	0	0
1 0 0 1	X	X	X	X	0	0	0	0	1	0	0	0	0	0	0

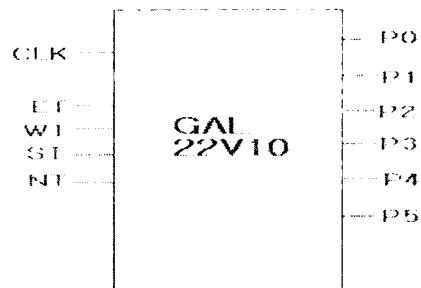


Fig 3-2 설계할 교통제어기의 블록다이어그램

4. GAL을 이용한 신호제어기 설계

GAL로 신호 제어를 설계하는데는 여러 가지 이점이 있다. Fig 3-2 에서와 같이 GAL을 이용한 설계는 GAL에 내장된 Flip Flop을 활용하므로 현재상태를 기억하는 상태 레지스터를 별도로 사용할 필요가 없이 GAL 하나만으로 설계할 수 있다.

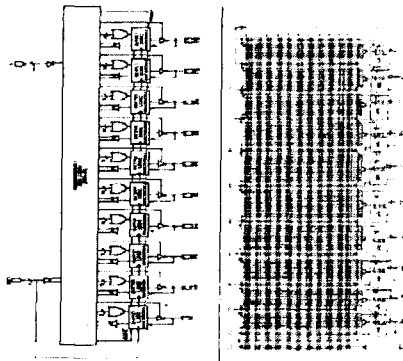


Fig3-3 GAL22V10의 블록 다이어그램

Table 3 에 의해 GAL22V10 으로 설계하기 위해 다음과 같은 부울식을 구한다  
부울식의 표현방법은 Mypld tool의 window version의 사용규칙을 따른다

- A= $A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D \cdot S \cdot N + A \cdot B \cdot C \cdot D \cdot S \cdot N + A \cdot B \cdot C \cdot D \cdot S \cdot N$ ;
- B= $A \cdot B \cdot C \cdot D \cdot E \cdot W + A \cdot B \cdot C \cdot D \cdot E \cdot W + A \cdot B \cdot C \cdot D \cdot E \cdot W + A \cdot B \cdot C \cdot D \cdot E \cdot W$ ;
- C= $A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D$ ;
- D= $A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D$ ;
- P5= $A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D$ ;
- P4= $A \cdot B \cdot C \cdot D \cdot E \cdot W + A \cdot B \cdot C \cdot D \cdot E \cdot W + A \cdot B \cdot C \cdot D \cdot E \cdot W + A \cdot B \cdot C \cdot D \cdot E \cdot W$ ;
- P3= $A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D$ ;
- P2= $A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D$ ;

- P1= $A \cdot B \cdot C \cdot D \cdot S \cdot N + A \cdot B \cdot C \cdot D \cdot S \cdot N + A \cdot B \cdot C \cdot D \cdot S \cdot N + A \cdot B \cdot C \cdot D \cdot S \cdot N$ ;
- P0= $A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot D$ ;

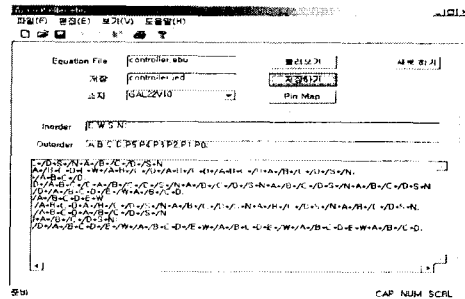


Fig 4-1 MYPLD에 부울식을 입력한 상태

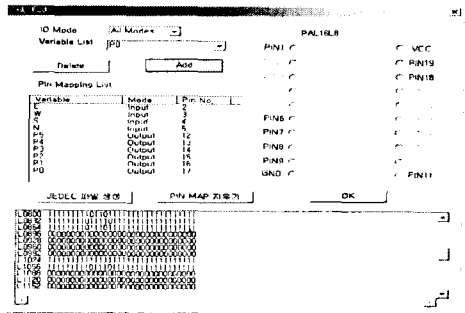


Fig 4-2 Pin assignment를 하고 JEDEC file을 생성한 상태

Fig 3-1과 같이 실제로 "OH" 부터 "9H" 까지 10개 이므로 상태를 표시하기 위해 A,B,C,D 4bit로 현재상태를 나타냈으며 센서 입력신호(WT, ET, ST, NT)그리고 다음상태로 A', B', C', D' 로 표시하였다. 그러나 실제 설계에서는 현재상태와 다음상태는 GAL내부에서 필요한 것이고 GAL소자의 Pin assignment에는 지정할 필요가 없다.

그리고 출력은 P5, P4, P3, P2, P1, P0 로 간단히 표시 하였다.

결국 신호제어기는 센서 입력신호에 따라 시간간격을 두고 출력 P가 적당히 표시되어야 한다. Table 4에 나타난 것과 같이 P5는

