

범용 제어기의 주변 소자 접속을 최적화하기 위한 전용 제어 회로의 구현 연구

•류 경식, 이 태훈, 정 기현, 김 용득
아주대학교 전자공학과

(A Study on an Implementation of the Custom Control Circuit for Optimizing the Interface of Peripheral Devices to General-Purpose Controllers)

•Kyeung Seek LEW, Tae Hoon LEE, Gi Hyun JUNG, Yong Deak KIM
Dept. of Electronic Eng. A-JOU University

ABSTRACT

This paper deals with the design scheme of the custom control circuit for optimizing the interface of peripheral devices to general-purpose controllers for the high speed digital system. When the various peripheral devices such as memory, I/O devices and buffers which operate at low speed are interfaced to the microprocessor which operates at high speed, inserting the proper wait state to the processor is required.

The proposed scheme designed with random logic may be applied to the high performance graphic system like the X-terminal. This circuit provides the flexibility and system independancy for the optimum digital system design.

I. 서론

근래의 디지털 시스템들은 매우 고속으로 동작하는 마이크로 프로세서를 내장하여 수십 MHz대의 주파수로 동작하는 것이 보편화 되어지고 있는 추세이며 이에 따라서 그 주변 소자들도 처리 속도가 계속 증가 되어지고 있다. 그러나 마이크로 프로세서에 접속이 되는 각종 소자들은 모두 처리 속도가 다르고 프로세서에 비하여 비교적 저속이기 때문에 프로세서가 접속된 주변 소자들을 가장 적은 시간으로 호출하기 위하여는 최소의 소자 호출을 위한 대기 시간을 프로세서에 인가 하여야만 최적으로 동작 시킬 수 있게된다. 그러므로 최적에 가까운 소자 접속을 위하여는 프로세서가 호출하고자 하는 주변소자들을 인식하여 그 소자에 알맞은 대기 상태를 인가하기 위한 제어 회로가 필요하게된다.

이러한 회로는 대부분 범용 논리 회로로 구성이 되지만 접속하고자 하는 소자가 많은 경우 게이트 수가 증가하여

전파 시간의 지연을 가져오게 되어 오히려 프로세서의 고속 동작을 저해할 우려도 있다. 이러한 단점을 극복하기 위하여 매우 고속으로 동작하는 대부분 디지털 시스템의 경우에는 PAL과 같은 프로그램형 논리 소자를 이용하여 시스템의 신뢰성을 높이고 아울러 주변 소자의 호출을 위한 최소의 대기 시간만을 할당 할 수가 있게 된다. 그러나 이러한 경우 시스템 설계시 확장성과 융통성을 제공하지 못하는 단점을 갖게되며 이의 구현을 위한 고가의 설계 프로그램과 이를 PAL로 구현하기 위한 프로그램 개발 장비가 필요하게 된다. 반면에 전용회로를 이용하여 하나의 MSI 급 집적회로로 구현 시켜 생산하는 경우 별도의 개발 장비나 프로그램 작성등의 번거러움 없이 프로세서의 버스 제어 회로에 간단하게 접속 함으로서 최적에 근사한 프로세서 대기 시간의 발생이 가능하게 된다. 그러나 PAL에 의한 회로에 대하여 효율이 조금 떨어지며 대기시간의 변경시 배선을 바꿔야 하는 문제가 대두된다.

본 연구에서는 마이크로 프로세서의 버스 제어시 최적의 제어를 구현하기 위한 회로를 PAL에 의하여 설계하고 이를 다시 게이트 단계의 전용 회로로 설계하여 이회로를 현재 개발 단계인 TMS34020 그래픽 프로세서를 이용하여 설계한 X-터미널용 고속 고해상도 그래픽 시스템에 장착하고 실험을 통하여 두 방식 모두 최적의 버스 제어를 구현할 수 있음을 확인 하였으며 이회로를 토대로 하여 하나의 TTL IC로 생산이 될 수 있으며 마이크로프로세서를 채용하는 비교적 간단한 디지털 시스템이나 시스템의 설계 및 개발과정의 회로에 적용하여 프로세서의 이용 효율을 높일 수 있게된다.

II. 주변 소자의 최적 운용 기법

디지털 시스템에서 고속으로 동작하는 하나의 마이크로 프로세서에 서로 다른 호출 시간을 갖는 여러 종류의 소자들을 접속 하고자 하는 경우 프로세서가 호출 하고자 하는

소자의 종류에 따른 적절한 대기 시간을 할당하여 프로세서를 대기 상태로 두어야 한다. 이를 위하여는 마이크로 프로세서의 동작 클럭에 정수배에 해당하는 호출 시간 그룹을 설정하고 접속하려는 소자들의 대기 시간에 따라 각 그룹으로 분류를 하고 해당 그룹의 소자 선택시 그 그룹에서 지정한 시간의 대기 시간을 발생 시키도록 하여야 한다. 프로세서의 클럭에 동기 되어 운용이 되는 디지털 시스템에 있어서 서로 다른 호출 시간을 갖는 7종류의 소자들을 50 MHz로 동작하는 프로세서에 접속하고자 하는 경우 각 소자들의 호출 시간과 프로세서에 인가 하여야 할 대기 사이클의 수는 표1의 경우처럼 주어질 수 있게된다.

<표1> 서로다른 호출 시간을 갖는 소자들의 예

소 자	호출 시간	대기 사이클 수	그룹
1	15 nsec	0	X
2	30 nsec	1	A
3	50 nsec	2	B
4	60 nsec	2	B
5	80 nsec	3	C
6	90 nsec	4	D
7	70 nsec	3	C

표 1에서 프로세서가 50 MHz의 주파수로 동작하고 프로세서 내부적으로 20 nsec의 클럭 사이클을 갖는다면 그러므로 20 nsec 이상의 호출 시간을 갖는 소자들은 매 20 nsec마다 하나의 대기 상태를 프로세서에 인가 하여야 만 올바르게 소자의 호출이 가능 하게 된다. 그러므로 15 nsec의 지연 시간을 갖는 1번 소자의 경우는 20 nsec 이하의 지연 시간이므로 별도의 프로세서 대기 시간이 필요하지 않게 된다. 그러나 일반적인 디지털 시스템에 접속된 소자는 프로세서의 동작 클럭에 동기되어 동작하게 되므로 동작 클럭의 정수배에 해당하는 호출 시간만이 가능하게 된다. 따라서 50 nsec의 호출 시간을 갖는 3번 소자와 60 nsec를 갖는 4번 소자의 경우 모두 B 그룹에 속하게 되어 같은 대기 상태를 인가하여야 한다. 따라서 위의 소자들을 A 부터 D 까지 20nsec단위로 그룹을 분할하는 경우 3번과 4번 소자 모두 B 그룹에 속하고 5번, 7번 소자는 C 그룹에 속하게 된다.

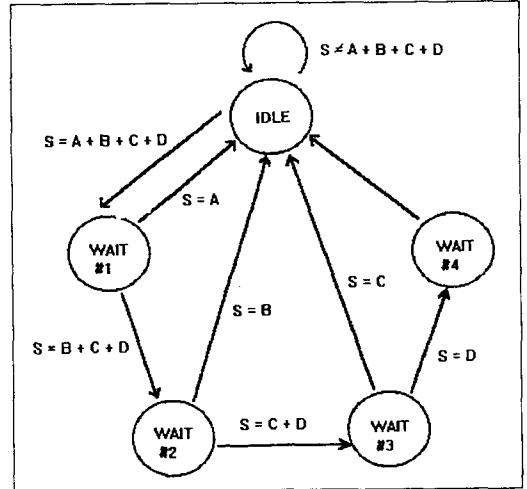
표1의 예와 같이 주변 소자가 접속이 되는 시스템의 경우 효율적인 대기 시간을 인가하는 제어 회로를 구현하기 위하여 일반적으로 범용 논리 소자를 이용하여 버스 제어 회로를 설계하거나 PAL과 같은 프로그램형 소자를 이용하여 설계하는데 이를 구현하기 위한 회로의 동작을 기술하면 그림1과 같다.

```

WHILE [IDLE]
  IF [A+B+C+D] THEN [WAIT1]
  IF [/(A+B+C+D)] THEN [IDLE]
WHILE [WAIT1]
  IF [B+C+D] THEN [WAIT2]
  IF [A] THEN [IDLE]
WHILE [WAIT2]
  IF [C+D] THEN [WAIT3]
  IF [B] THEN [IDLE]
WHILE [WAIT3]
  IF [D] THEN [WAIT4]
  IF [C] THEN [IDLE]
WHILE [WAIT4]
  IF [D] THEN [IDLE]

```

(가) 의사 프로그램 방식을 이용한 회로의 기술



(나) 상태 변이 다이어그램을 이용한 회로의 기술

<그림1> 대기시간 인가를 위한 제어회로의 기술

그림 1에서 나타난 바와같이 프로세서에 소자의 해당 그룹에 따른 적절한 대기 시간을 인가하기 위하여는 버스 상태 제어 회로부에서 프로세서가 현재 어느 소자를 호출하려는지 파악하고 대기 상태를 인가 회로부에서는 그림1(나)의 상태도에 의한 적절한 대기 상태를 프로세서에 인가하여야 한다. 따라서 이와 같은 시스템의 경우에 있어서 모든 주변소자들은 프로세서가 20nsec를 기준으로 동작하므로 20 nsec의 정수배에 해당하는 시간으로만 주변 소자를 호출할 수 있게 된다. 그러므로 프로세서 동작 주파수를 MCLK, 그 주기를 tCLK이라 할때 tCLK의 정수배가 아닌 호출 시간을 갖는 주변 소자의 경우는 식(1)으로부터 구할 수 있는 정수 N을 구함으로써 해당 소자의 최적 호출을 위한 프로세서 대기 클럭수를 구하게 된다. 이때, X는 소자가 갖는 호출시간, N은 구하고자 하는 프로세서 대기 사이클 수이다.

$$tCLK * N < X \leq tCLK * (N+1) \text{ ----- 식(1)}$$

따라서 접속하고자 하는 모든 소자들이 식(1)의 조건을

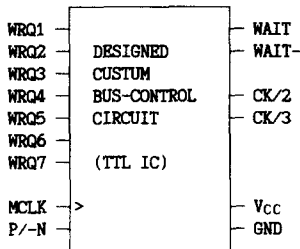
만족 하는 대기 클럭을 프로세서에 할당하도록 제어 시스템을 설계한다면 이 시스템은 주변 소자를 최적으로 호출하는 최적 버스 제어를 구현하게 되며 이로써 시스템의 운용 효율이 극대화되며 프로세서의 휴지시간을 최소화되어 고속 디지털 시스템의 실현이 가능하게 된다.

예로서 50 MHz 로 고속 동작하는 디지털 시스템에 있어서는 접속된 소자가 호출시 불과 10 nsec 정도의 시간 손실이 발생한다 하더라도 2번의 호출로 한 클럭의 손실이 발생하며 이 소자의 호출 빈도가 많거나 호출 시간이 프로세서의 동작 클럭에 비하여 그다지 크지않은 경우에는 시스템의 효율적인 프로세서 운용에 상당한 손실이 발생하게 된다. 그러나 접속 소자의 호출 시간이 클럭 시간의 정수배가 아닌 경우에 발생하는 경우의 시간 손실은 클럭에 동기되어 운용되는 디지털 시스템에 있어서는 불가피한 것이다.

III. 전용 회로의 구성과 설계

설계한 전용 회로는 최적 운용을 구현하기 위한 조건식인 식1을 만족하도록 설계되어 있으며 또한 시스템의 설계자가 버스제어 회로와 접속 시키기가 매우 용이하도록 고려되었으며 별도의 개발 환경이나 별다른 회로 설계가 필요하지 않도록 설계하였다.

그림2에는 실제 설계한 전용회로가 MSI급의 IC로 생산될 경우의 접속 핀들을 나타내었으며 표2에는 이 회로의 각 핀들의 기능을 설명하였다.



<그림 2> 전용 집적회로의 핀 배치

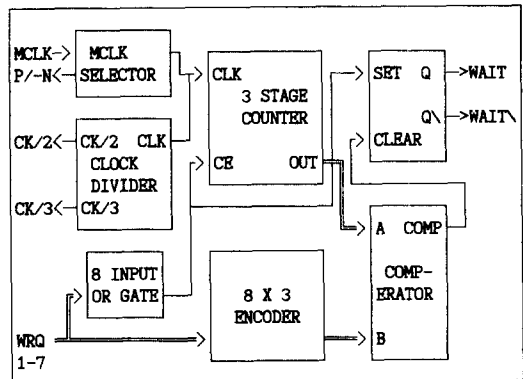
설계한 전용 IC 회로의 내부 주요 구성을 블럭도로 나타내면 그림3과 같다.

그림3의 회로에서 프로세서의 대기 상태 인가를 요구하는 WRQ1~WRQ7의 입력중 하나가 발생하면 WAIT 신호의 출력단 플립플롭이 세트 되어 이 단자에 접속된 프로세서를 대기상태로 만들며 동시에 8 입력 OR 회로에 의하여 3단 계수기를 클리어 시켜 출력을 초기화 하게된다. 이후 3단 계수기는 P/-N의 구성에 따라 선택 되는 프로세서의 기준 동작 클럭 MCLK의 클럭수를 계수하기 시작하며 8*3인코더는 요구한 WRQ1-7의 대기상태의 요구신호에 의하여 입력이 결정되고 이를 3비트의 2진 코드로 생성 시킨다. 이 3비트의

인코더 출력 코드는 비교기의 입력이 되어 계수기의 최대 계수 클럭수를 제한한다. 따라서 계수기의 클럭계수 값과 인코더의 출력코드의 값이 같게 되면 비교기에 의하여 COMP 신호가 활성화되고 이 신호가 종단의 플립플롭을 직접 클리어 시켜 WAIT신호를 다시 LOW상태로 하여 프로세서를 대기 상태에서 해제 시킨다. 이회로에서는 입력되는 WRQ 요구 신호 단자를 다시 비활성화 시키는 회로부가 없는데 이는 PAL 등에 의해 설계하는 회로와 마찬가지로 거의 대부분의 프로세서들과 그 버스 제어 회로가 WAIT 신호의 해체에 의하여 WRQ 신호를 자체적으로 비활성화 시키기 때문이다. 따라서 시스템을 설계하는 경우 버스 디코딩 회로의 각 소자 선택 신호선들을 그룹별로 NAND 게이트로 묶고 이를 WRQ 신호에 접속하고 프로세서가 이용하는 주변 소자 운용 클럭을 MCLK에 접속하며 P/-N을 프로세서의 종류에 맞게 설정한후 프로세서의 WAIT혹은 READY 신호에 WAIT나 -WAIT를 접속하기만 하면 된다.

<표 2> 외부 접속 신호들의 기능

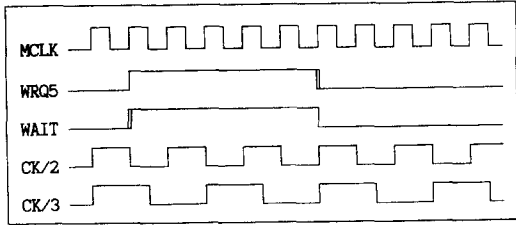
신호명	기능
WRQ1 -WRQ7	대기클럭 요구신호 입력단자이며 기준클럭에 대하여 1 ~ 7개중 하나의 대기클럭 요구
MCLK	전용회로의 대기 상태 발생을 위한 기준클럭
P/N-	MCLK의 유효레벨을 규정하며 프로세서의 유효레벨과 일치 시킨다. HIGH이면 MCLK의 상승에서 LOW이면 MCLK의 하강에서 동기
CK/2	MCLK를 2분주한 신호를 출력하는 신호
CK/3	MCLK를 3분주한 신호를 출력하는 신호
WAIT	프로세서의 대기 상태인가 신호, 정논리
WAIT-	WAIT 신호의 부논리 신호



<그림 3> 설계된 전용회로의 블럭 구성도

설계한 전용회로의 P/N-신호를 플립 시켜 MCLK의 상승 시간에 동기를 맞추고 WRQ5가 요구된것으로 가정하여 회로

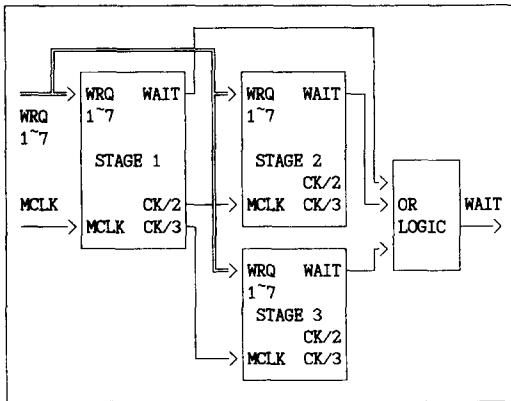
를 동작시킬때 각 신호들의 타이밍 다이어그램은 그림 4와 같이 나타난다.



<그림 4> 전용회로의 동작 타이밍 다이어그램

그림 4에서 WRQ5의 선택으로 WAIT 신호가 활성화 되고 MCLK의 상승 시점에서 계수를 시작하여 5개의 MCLK신호를 계수하고 난 후에 WAIT 신호를 해제 시킨다. 그후 버스 디코딩 회로에서 WAIT 신호의 비활성 상태를 인식하고 WRQ5신호를 비활성으로 만든다.

설계한 회로는 MSI급의 TTL IC로 구현이 가능하도록 하기 위하여 핀수를 줄이고자 1에서 7개 까지의 대기클럭 인가만을 지원하고 있는데 대부분의 디지털 시스템의 경우에는 이 정도의 기능이면 충분히 활용 할 수가 있게된다. 그러나 만약 8개 이상의 많은 대기 시간을 요구하는 신호가 있는경우가 발생할 수가 있는데 이경우는 전용회로를 다단으로 접속 하여 사용하면 된다. 이러한 다단접속을 위하여 CK/2 신호와 CK/3 신호를 제공하고 있다. CK/2 신호와 CK/3 신호는 단순히 입력되는 MCLK를 각각 1/2, 1/3으로 분주하여 출력하며 첫째단의 CK/2를 둘째단의 MCLK에 접속하고, CK/3 단자를 셋째단에 MCLK 입력으로 사용하여 대기클럭의 요구선을 확장한다. 그림 5에 8개 이상의 대기 클럭을 얻기 위하여 전용회로 3개를 3단으로 구성된 회로를 불러다이어그램으로 나타내었다.



<그림 5> 전용회로 3개를 사용한 2단 접속 회로

그림5의 회로에서 첫째단의 MCLK을 분주한 신호인 CK/2, CK/3이 다음단의 MCLK가 되므로 이러한 구성에서 제공되는 클럭의 지원가능 갯수는 1단에서 1~7개의 MCLK,

CK/2를 접속한 2단에서 2, 4, 6, 8, 10, 12, 14개의 MCLK, CK/3을 접속한 3단에서 3, 6, 9, 12, 15, 18, 21개의 MCLK로 주어지며 그 이상의 대기 클럭이 필요한 경우는 다시 2, 3단의 CK/2, CK/3 신호를 이용하여 제조합으로 생성하거나 직접 1단의 MCLK를 필요한 갯수만큼으로 분주하여 사용할 수가 있게된다. 그러나 11, 13, 17, 19, 23, 27등과 같은 10 이상 숫수의 대기 클럭 요구가 필요한 경우는 전용회로의 다단 접속만으로는 제공할 수가 없게되며 이러한 경우는 필요한 대기 클럭 보다 하나의 대기클럭을 더하여 호출 하여야 한다. 그러므로 각각 WRQ12, 14, 18, 20, 24, 28의 신호를 이용 하여야 하고 이로 인한 1클럭씩의 손실은 전용회로의 이용에서는 불가피하게 된다. 그러나 이러한 문제는 10개 이상의 대기를 요하는 매우 저속의 소자를 접속할 때에 발생하며 대개의 디지털 시스템이 8개 이상의 대기를 요하지 않으므로 시스템의 최적 운용의 관점에서 볼때 큰 문제점을 야기하지 않는다.

전용회로의 다단 접속으로 제공하지 못하는 10 이상 숫수개의 대기클럭이 필요한 경우에 발생하는 시간 손실율은 실제 소자의 호출에 필요한 시간에 대한 추가 시간의 비로 주어지며 식(2)와 같이 주어진다. 식(2)에서 N은 10이상의 숫수이며 t_{CLK} 는 MCLK의 주기이다.

$$\begin{aligned} \text{시간 손실율}(x) &= \frac{[MCLK * (N+1)] - [MCLK * N]}{(MCLK * N+1)} * 100 \\ &= \frac{1}{N+1} * 100 \quad \text{----- 식(2)} \end{aligned}$$

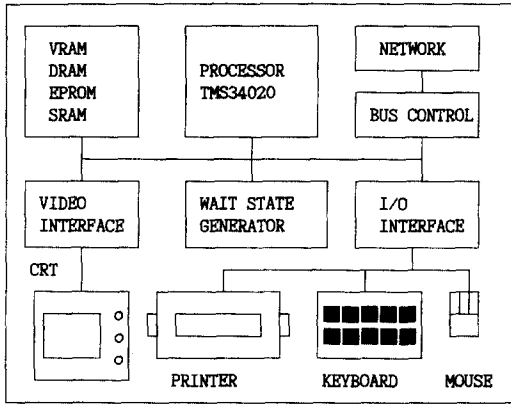
식(2)에서 보는바와 같이 11, 13, 17, 19, 21개의 대기 클럭에 대하여 시간 손실율은 각각 8.33%, 7.14%, 5.00%, 4.54%로서 프로세서의 주파수에 대한 상대적인 시간 손실율의 크기는 주파수에 무관하게 같으나 절대적인 시간 손실의 크기는 프로세서의 동작클럭의 주기와 같으므로 주파수가 증가 할 수록 감소하게 된다.

IV. 고속 그래픽 시스템 설계에의 적용 실험 및 고찰

본 연구에서 설계한 전용 회로를 이용하여 X-윈도우용 그래픽 터미널인 X-터미널을 구현하기 위한 고속 고해상도의 그래픽 시스템의 설계시에 적용하였다. 그림 6에 TI사의 32 비트 그래픽 전용 프로세서인 TMS34020을 이용하여 설계한 그래픽 시스템의 블록 구성도를 나타내었다.

구현한 시스템은 X-터미널의 기본 사양인 마우스와 키이보드, 프린터 및 모니터를 접속하였으며 프로세서에 접속되어 있는 주변 소자들은 크게 VRAM, DRAM, EPROM, SRAM, 직렬 포트, 병렬 포트등이다. 32.4MHz로 동작하는 TMS34020 프로세서에 이들 소자들의 호출을 위하여 인가하여야 하는

대기 클럭수에 의한 그룹 분할은 표3과 같이 주어진다.



<그림 6> 그래픽 처리 시스템의 구성도

<표 3> 소자들의 특성과 해당 그룹

그룹	대기클럭	접속 소자
A	0 MCLK	DRAM, VRAM, SRAM, INTERNAL REGISTER
B	1 MCLK	STATUS LED
C	2 MCLK	PALETTE, EPROM, EEPROM
D	3 MCLK	KEYBOARD, MOUSE, PRINTER
E	4 MCLK	NETWORK

표3에서 주어진 소자들을 접속하기 위한 버스 제어 회로를 위하여 R 타입의 PAL 20R4를 이용하여 최적화를 꾀한 회로와 본 연구에서 설계한 전용회로를 이용한 회로로 각각 구현하였다. 그림 7은 PAL 소자를 이용한 경우의 ABEL 형식 프로그램의 일부를 나타낸다.

```

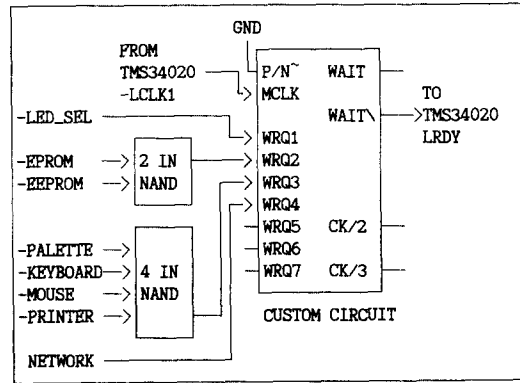
"Equations"
S1 = !LED_CS;
S2 = !Palette # !Keyboard # !Mouse # !Printer;
S3 = !EPROM # !EEPROM;
S4 = !Network;

"Wait state definition"
State IDLE:IF (S1 # S2 # S3 # S4)
    THEN WAIT1
    ELSE IDLE;
State WAIT1:IF (S1) THEN IDLE
    ELSE WAIT2;
State WAIT2:IF (S2) THEN IDLE
    ELSE WAIT3;
State WAIT3:IF (S3) THEN IDLE
    ELSE WAIT4;
State WAIT4:IF (S4) THEN IDLE
    ELSE IDLE;
    
```

<그림 7> PAL 회로로 구현한 회로의 프로그램

그림 7에서 보는 바와 같이 PAL을 이용한 대부분의 회로는 프로그램이나 직접 비트패턴에 의한 퓨즈맵을 작성하

여 설계하므로 하드웨어적으로는 간편하나 이를 구현하기 위한 개발장비와 환경이 필요하다. 그림 8은 설계한 전용 회로를 이용하여 대기 상태를 발생 시키기 위한 경우 TMS34020 그래픽 프로세서와 주변 소자들의 회로 접속을 나타낸다.



<그림 8> 전용회로를 이용한 경우의 접속

그림 8의 경우 표3에서 정의한 신호들은 X-터미널 그래픽 시스템의 버스 디코딩 회로에서 각 소자의 고유 주소를 이용하여 각각의 I/O 선택 신호를 출력한다. 이 신호는 해당 소자의 -CE 활성화 신호로 이용이 되며 동시에 그룹별로 NAND 게이트에 입력이 되어 이것이 WR신호를 활성화 시키게 된다. 기존 클럭인 MCLK는 TMS34020의 -LCLK를 이용하였으며 -LCLK의 부논리를 충족하기 위하여 P/N을 논리 0으로 하였다. -WAIT신호는 TMS34020의 LRDY 신호에 접속하였는데 LRDY 신호는 논리 1에서 프로세서가 정상으로 동작하며 논리 0에서는 대기 상태에 들어가게 된다.

이와같이 PAL로 구현한 방식은 프로그램의 변경에 의하여 내부 변경이 가능하므로 손실이 전혀 발생하지 않으며 설계한 전용회로를 이용한 경우도 7개 이하의 대기 클럭을 요구하는 소자들만 접속이 되어있으므로 시간의 손실이 전혀 문제되지 않았다. 표3에서 그룹으로 할당된 소자들이 모두 식(1)을 만족하므로 소자의 호출시 항상 손실이 없는 최적의 상태로 프로세서를 운용 함을 확인 할 수 있다.

V. 결론

본 연구에서 구현된 전용회로는 논의한 바와 같이 프로세서에 접속이 되는 주변 소자제어를 최적의 시간으로 호출이 가능하도록 하기 위하여 설계되었으며 사용자가 설계하고자 하는 시스템의 버스 디코딩 회로에 적용이 용이하도록 되어 있다. 이 전용 회로를 이용하여 X-터미널에 접속이 되는 고해상도의 그래픽 시스템에 접속하여 실험한 결과 PAL로 구현한 경우와 비교하여 전혀 문제없이 최적의 시간으로

로 프로세서를 운용함을 확인 하였다.

설계된 전용회로에서는 10 이상의 숫수개에 대한 대기 상태의 발생에서는 매번 호출시마다 1클럭씩의 손실을 가져 오지만 이러한 대기 시간을 요구하는 주변 소자는 극히 드물며 전체 운용 시간에 대한 그 손실율은 매우 미소함을 이미 위의 논의에서 검토 하였다. 그러나 PAL등에 의하여 최적으로 운용되도록 설계된 회로보다 본 전용회로로 설계된 경우가 그접속이나 구현이 매우 간단하고 PAL등으로 구현하기 위한 고가의 장비나 프로그램 없이도 효율적인 버스제어 효과를 얻을 수 있다. 따라서 이 회로를 토대로 MSI급의 TTL IC로 구현하는 경우 개발 단계의 시스템 설계나 비교적 많은 주변 소자를 이용하지 않는 간단한 시스템에 응용하여 마이크로 프로세서의 최적 운용을 실현 할 수가 있게된다.

앞으로 본 연구에서 구현한 전용회로에 접속 되는 소자들의 대기 시간 요구 범위가 광범위한 경우 다단 접속을 해야 하는 문제점과 10 이상 숫수개의 대기 클럭 발생이 불가한 점들을 해결하기 위하여 이 회로를 기본적으로 응용한 프로그램형 LSI소자 형태로 설계하여 더욱 사용자의 접속이 간편하도록 하는 한편 보다 신뢰성있는 최적의 제어 상태를 얻을 수 있도록 계속 연구가 진행 되어야 한다.

참고 문헌

- [1] 김 용득 "X-터미널의 구현을 위한 고해상도 그래픽 시스템의 개발", 대한전자공학회 추계 학술 발표대회 논문집, Vol 14, No 2, pP 428-432, 1991.11
- [2] 김 용득 "X-윈도우용 그래픽 터미널의 접속 기술 개발에 관한 연구", (주)인텔 최종 연구 보고서, 1992.2
- [3] Foley, Van Dam and Hughes " Computer Graphics " second edition, Addison-Wesley, 1990
- [4] Martin Bolton " Digital System Design with Programmable Logic", Addison-Wesley, 1990
- [5] Angel E. Socarras and Robert S. Cooper " Anatomy of an X-terminal " IEEE Spectrum, 1991.3
- [6] James Buchanan " CMOS/TTL Digital System Design ", Siliconix INC, 1989
- [7] Elizer Sternheim " Digital Design with Verilog HDL " Design Automation Series, 1990
- [8] D. Del Corso "Microcomputer Buses and links" Academic Press, 1989