【 기술 노트 16 】

마이크로프로세서에서 병렬 입출력 회로의 올바른 설계

마이크로프로세서에서 병렬 입출력 장치를 설계할 때는 CPU와 입출력 장치의 동작 특성 및 요구 타이밍에 따라 세심한 설계가 필요하다. 흔히들 디지털 신호의 논리 레벨만을 고려하여 회로를 설계하거나, 여기에 고작해야 액세스 타임(access time)을 고려하여 적절히 웨이트 사이클(wait cycle)을 부여하는 정도로 입출력 인터페이스를 처리하는 것이 보통이지만, 이렇게 피상적으로 인터페이스 회로를 설계하면 계절이나 환경의 변화에 따른 반도체 온도 특성에 의하여 회로의 동작이 오락가락 불안정해지거나 또는 어느 반도체 회사의 논리 소자를 사용하느냐에 따라 동작의 안정성이 영향을 받는 경우가 발생할 수 있게 된다.

이러한 문제가 발생하는 것은 모두 마이크로프로세서와 주변장치의 요구 타이밍을 고려하지 않고 인터페이스 회로를 설계하였기 때문이다. 특정한 소자의 동작 지연 타이밍값에 의해서가 아니라 CPU의 클럭 주기를 기준으로 정확한 동작이 이루어지도록 시스템을 설계하여야만 이런 문제를 근원적으로 방지할 수 있다. 즉, 액세스 타임은 크게 보아 CPU의 전체적인 외부 액세스 동작 시간과 74LS, 74HC, 74HCT 등과 같이 인터페이스 회로에 사용되는 소자의 계열에 의하여 결정되지만, 같은 74LS 계열의 디지털 소자들을 사용하더라도 이 소자들끼리 미세한 특성 차이나 반도체 제조회사에 따른 약간의 동작 타이밍 또는 전압 레벨의 차이에 의하여 인터페이스 회로가 영향을 받지 않도록 하려면 CPU의 기본 동작을 기준으로 정확한 동작이 이루어지도록 회로를 설계해야 한다는 것이다.

마이크로콘트롤러에는 많은 병렬 포트를 내장하고 있어서 이를 직접 사용하는 경우가 많지만 필요하다면 여기에서도 추가로 외부에 병렬 인터페이스를 접속하여 사용하는 경우가 있으며, CPU마다 서로 동작 타이밍이 다르므로 이러한 회로의 인터페이스 방법이 모두 같지는 않으나 기본 원리는 비슷하므로 개념만 분명히 알아두면 어디서나 별 어려움이 없이 설계를 수행할 수 있다.

병렬 인터페이스에서 출력단의 라이트(write) 회로는 일반적으로 래치(latch) 기능을 필요로 하므로 CPU에서 출력되는 데이터와 래치에 인가되는 클럭 신호의 관계가 중요하며, 입력단의 리드(read) 회로는 3스테이트 버퍼(3-state buffer)에 의하여 외부 입력 데이터를 CPU의 데이터 버스에 접속하여 주므로 이 버퍼가 열리는 시간과 CPU가 데이터 버스를 실제로 읽어들이는 순간의 타이밍이 중요하다. 이처럼 병렬 출력 회로와 입력 회로는 그 기본 원리부터 다르기 때문에 여기서는 이를 2가지로 나누어 설명하기로 한다.

1. 출력단 래치 회로의 설계

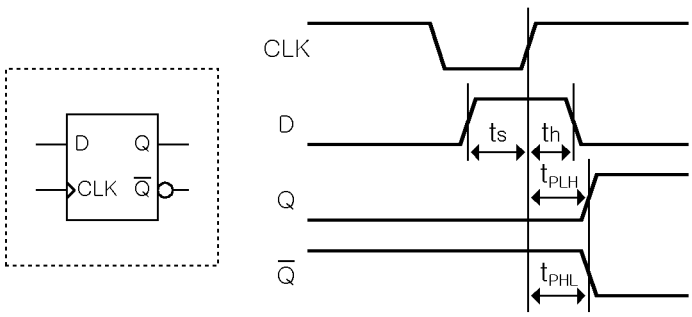
마이크로프로세서에 인터페이스하는 병렬 출력회로는 거의 대부분 래치 회로를 사용한다. 이것은 CPU가 출력 명령으로 외부 번지에 라이트하는 데이터는 그 명령이 실행되는 불

과 수십 내지 수백 [ns] 동안만 잠깐 데이터 버스에 나타났다가 사라져 버리므로 이를 래치에 기억시켜 두지 않으면 출력장치에 지속적으로 원하는 데이터를 출력할 수 없게 되기 때문이다. 예를 들어 LED를 점등하기 위한 출력회로라면 래치를 사용하지 않을 경우 LED가 점등되는 시간이 너무 짧아서 사람에게는 LED가 켜지는 것이 거의 보이지도 않게 된다.

이러한 래치의 기본 동작은 <그림 1>과 같다. 즉, 래치에 입력되는 데이터가 클럭 신호의 상승 에지(rising edge)에서 래치에 올바르게 저장되어 출력신호 Q 및 $\overline{Q}$에 나타나도록 하려면 적어도 클럭 신호의 상승 에지보다 셋업 타임(setup time) t_s만큼 이전부터 데이터 신호 D가 안정되게 입력되고 있어야 하며, 클럭 신호가 상승한 이후에도 최소한 홀드 타임(hold time) t_h만큼이 지나도록 D 신호가 같은 상태로 안정되게 유지되어야 한다. 만약, 이러한 조건이 지켜지지 못한다면 D값이 올바르게 래치되는 동작을 보장할 수 없게 된다. 이 그림에서 t_{PLH}와 t_{PHL}은 각각 클럭 신호에 의하여 래치된 출력값이 외부로 나타나기까지의 H→L 및 L→H 전달지연시간(high-to-low or low-to-high propagation delay time)이다.

이를 마이크로프로세서의 병렬 출력 래치회로에 대비시켜 보면 데이터 신호 D는 CPU의 데이터 버스에 해당하고 CLK 신호는 어드레스 디코더에 의한 칩선택 신호($\overline{CS}$)가 되는데, 이중에서 특히 칩선택 신호는 CPU의 동작에 맞추어 상당히 세심하게 타이밍을 고려하여 설계하지 않으면 이러한 래치 소자의 타이밍 요구 조건을 만족하지 못하게 될 수가 있다.

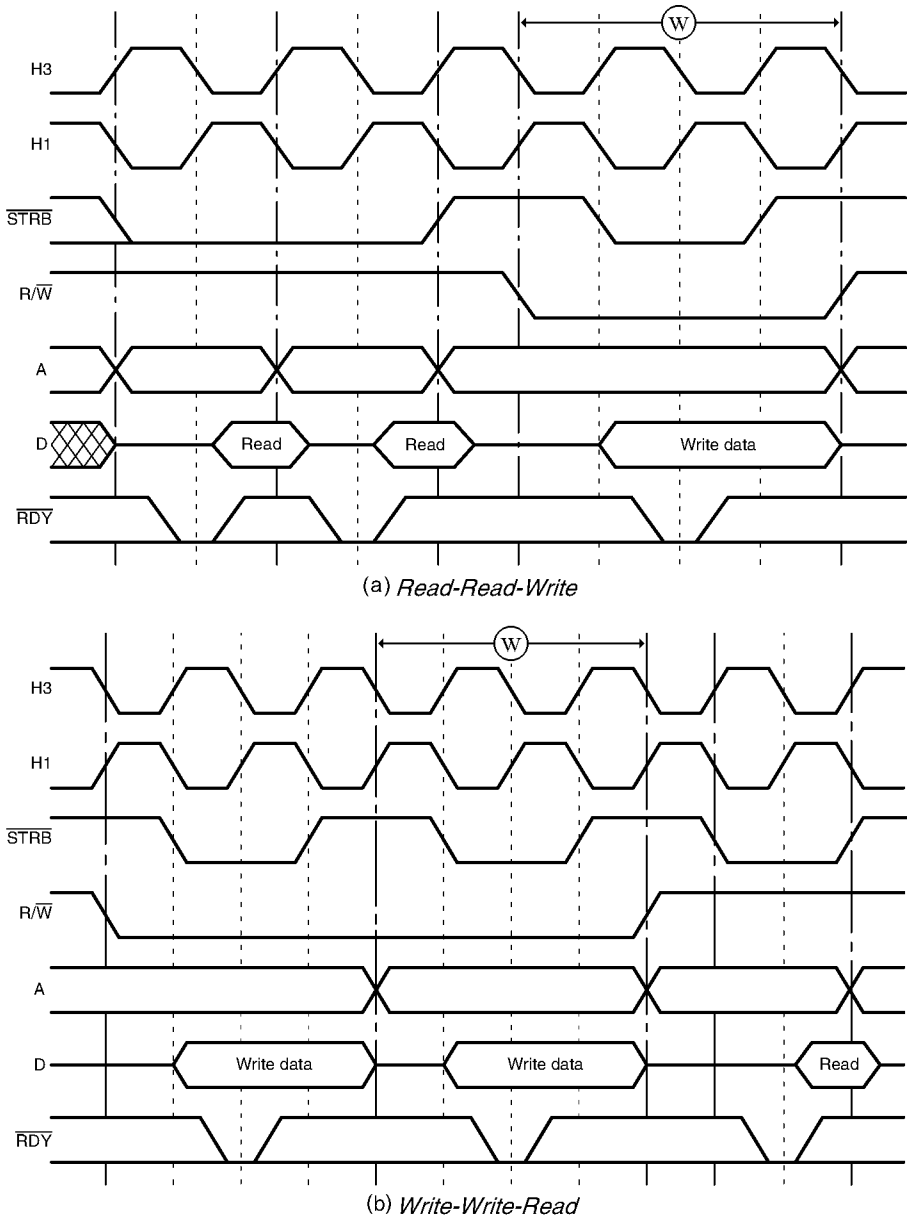
칩선택 신호의 기능이나 논리 레벨만을 생각하면 어드레스값만을 가지고 디코더 회로에 의하여 이를 만들어도 아무 문제가 없으므로, 흔히들 이 디코더의 출력과 CPU의 $\overline{WR}$ 신호를 부논리로 AND(정논리로 OR)하여 $\overline{CS}$ 신호를 만드는 것을 보고도 이는 출력할 때만 사용하는 것이므로 $\overline{WR}$ 신호를 함께 묶어서 사용하였으려니 하고 대수롭지 않게 생각하게 된다. 그러나, 이는 그러한 의미도 물론 있을 수 있지만 대부분은 그보다 더 중요한 타이밍 문제 때문에 사용하는 것이라는 것을 이해해야 한다. 어드레스 디코더에 의하여 만들어진 신호가 래치에서 요구하는 타이밍을 만족하지 못하므로 이를 $\overline{WR}$ 신호로 타이밍을 조절하여 $\overline{CS}$ 신호를 발생시킴으로써 올바른 래치 동작이 가능하도록 해주는 것이다.



<그림 1> 기본적인 래치의 동작 파형

▣ TMS320C31 DSP의 경우

일반적으로 고속 동작을 중시하는 DSP에서는 하드웨어적으로 1클럭 주기에 외부 액세스가 수행되므로, 기본적으로 4클럭에 이를 수행하는 80C196KC나 12클럭에 동작하는 8051 등의 마이크로콘트롤러에 비하여 상당히 인터페이스 회로의 설계가 상당히 까다로와진다. 그러나, 아마도 이러한 이유 때문인지 TMS320C31 DSP에서는 리드 동작은 1클럭에 수행하지만, 라이트 동작은 2클럭에 수행하도록 만들어져 있으므로 이러한 하드웨어 설계의 어려움이 다소 완화된다.



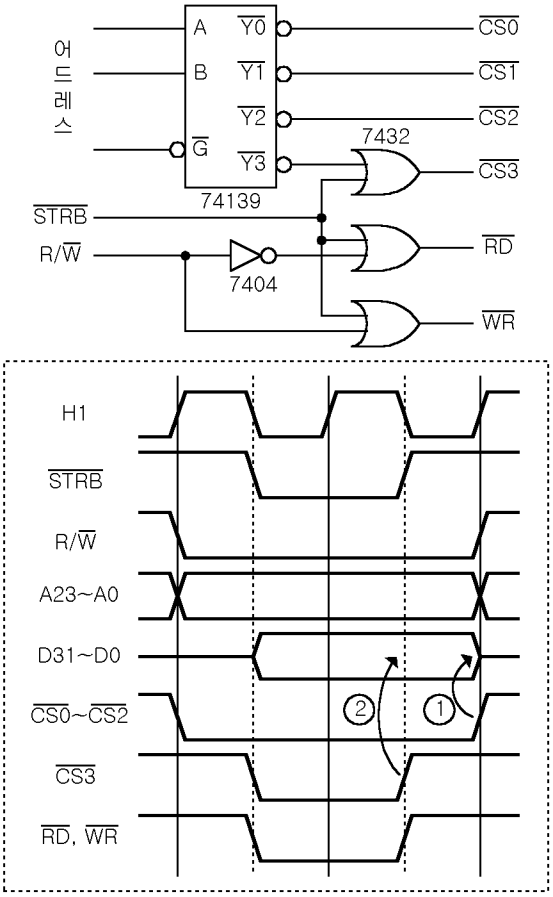
<그림 2> TMS320C31의 출력 라이트 동작 타이밍

TMS320C31 DSP의 외부 메모리 또는 I/O에 대한 라이트 동작은 <그림 2>와 같다. 그림 (a)에서 보는 바와 같이 리드 동작에 이어지는 라이트 동작과 (b)처럼 라이트가 연속될 때의 동작은 전체적으로 약간 다르지만, Ⓦ로 표시한 부분의 2클럭 동안에 이루어지는 순수한 라이트 동작은 완전히 동일하다. 즉, 이 H1 클럭을 기준으로 2클럭 주기동안에 라이트가 이루어진다고 생각할 수 있다.

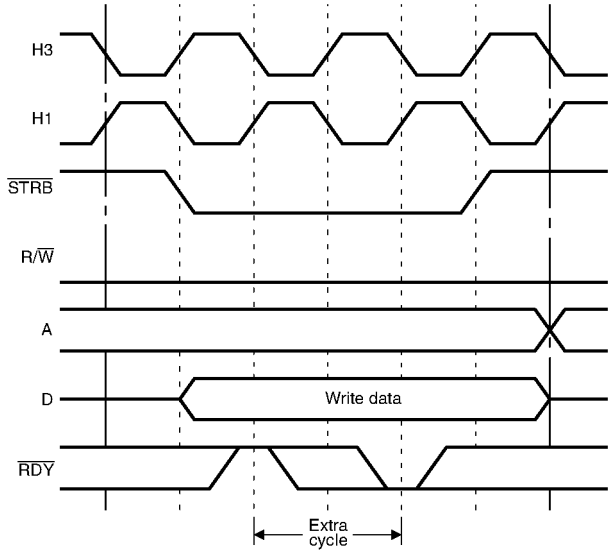
그림에서 수직으로 표시한 점선처럼 이 라이트 동작 사이클을 4개의 구간으로 나누어서 생각한다면, R/$\overline{W}$ 신호 및 어드레스 버스는 전체 4개의 구간동안에 유효하게 출력되고, $\overline{STRB}$ 신호는 중간 부분의 2개 구간만 유효하며, 출력되는 데이터 버스는 후반부 3개의 구간에만 유효하다는 것에 주목하여야 한다. 특히 여기서 관심있게 보아야 하는 것은 다른 마이크로프로세서에서의 일반적인 경우와는 달리 R/$\overline{W}$ 신호 및 어드레스 버스 동작 시간이 완전히 동일하다는 것이다. 이는 하드웨어 설계에 상당히 문제를 야기시키므로 이를 해결할 수 있도록 이 DSP에서는 $\overline{STRB}$ 신호를 가지고 있어서, R/$\overline{W}$ 신호와 $\overline{STRB}$ 신호를 부논리로 AND하면 다른 마이크로프로세서의 $\overline{WR}$에 해당하는 신호를 만들 수 있으며, R/$\overline{W}$ 신호를 반전시켜 $\overline{STRB}$ 신호와 부논리로 AND하면 다른 마이크로프로세서의 $\overline{RD}$에 해당하는 신호를 만들 수 있게 된다.

즉, 어드레스 버스만을 가지고 디코딩된 칩선택 신호는 이 4개의 구간동안 L상태를 유지하다가 라이트 동작의 마지막 순간에 상승 에지가 되므로, 데이터 버스가 셋업 타임은 충분하지만 자칫하면 홀드 타임을 만족하지 못하게 된다. 더구나, 데이터 버스는 DSP에서 출력된 그대로인데 비하여 어드레스 디코더를 거쳐서 만들어진 칩선택 신호는 어드레스 디코더에서의 시간지연 때문에 데이터 버스보다 약간 늦게 상승 에지를 나타내므로 이때는 이미 데이터 버스의 동작이 종료된 이후가 되어 정상적인 라이트 동작이 수행될 가능성은 거의 없다. 따라서, TMS320C31 DSP에서 출력 래치 회로를 설계하려면 이상의 사항을 고려하여 어드레스 디코더의 출력을 그냥 래치의 클럭 신호로 사용하지 말고, 이를 $\overline{STRB}$ 신호와 결합하든지 또는 $\overline{STRB}$ 신호를 사용하여 만든 $\overline{WR}$ 신호와 결합하여야 한다.

이를 실제로 구현하는 회로와 동작 파형으로 설명하면 <그림 3>과 같다. 즉, 어드레스 신호만을 가지고 74139 어드레스 디코더로 디코딩하여 만들어진 칩선택 신호 $\overline{CS0}$ ~ $\overline{CS2}$는 그 타이밍이 어드레스 신호와 같고, 여기에 어드레스 디코더에서의 동작 지연시간까지를 고려하면 그림에서 ①로 표시한 것처럼 이 신호의 상승 에지에서 데이터를 올바르게 래치할 수 없다. 그러나, 어드레스 디코더의 출력과 $\overline{STRB}$ 신호를 부논리로 AND하여 만든 신호 $\overline{CS3}$는 그림에서 ②로 표시한 것처럼 항상 데이터 버스의 종료보다 ½클럭 주기(시스템 클럭으로는 1클럭 주기)만큼 먼저 상승 에지를 나타내므로 래치에서 요구하는 충분한 홀드 타임을 갖게 되어 정확하고 안정된 래치 동작을 수행한다.



<그림 3> TMS320C31에서 올바른 칩선택 신호의 발생



<그림 4> 1웨이트가 주어진 TMS320C31의 출력 라이트 동작 타이밍

이러한 사항은 74LS574와 같은 8비트 D플립플롭을 사용하여 병렬출력장치를 인터페이스 하는 경우나 이와 유사한 동작을 필요로 하는 LCD 모듈에서도 반드시 필요하다. 그러나, 칩선택 신호와 $\overline{RD}$ 및 $\overline{WR}$ 신호를 모두 필요로 하는 RAM, ROM 등의 반도체 메모리 소자나 82C55A, 래치를 내장하고 있는 D/A 컨버터와 같은 주변 LSI를 인터페이스하는 경우에는 절대로 <그림 3>에서와 같이 만든 $\overline{CS3}$ 신호가 아니라 $\overline{CS0}$ ~ $\overline{CS2}$의 신호를 칩선택 신호로 사용하여야 하며, 여기에 추가로 $\overline{STRB}$를 이용하여 만든 $\overline{RD}$ 및 $\overline{WR}$ 신호를 사용해야 한다는 것을 잊지 말아야 한다. 그런 소자들은 칩선택 신호에 의하여 전체의 동작이 인에이블(enable)되고 이보다 폭이 짧은 $\overline{RD}$, $\overline{WR}$ 신호에 의하여 이러한 문제가 내부에서 처리되도록 설계되어 있기 때문이다.

이상과 같은 내용은 클럭 신호를 기준으로 동작하므로 클럭 주파수가 상당히 달라지더라도 문제없이 동작하며, 웨이트 사이클을 갖는 느린 소자의 리드/라이트 동작에서도 마찬가지로 적용된다. 예를 들어 <그림 4>에서와 같이 1개의 웨이트 사이클이 추가되는 경우에는 $\overline{RDY}$ 신호의 상태에 따라 결과적으로 이 1사이클(extra cycle)이 $\overline{STRB}$ 신호의 정중앙 부분에 추가되어 총 3개의 H1 클럭 주기로 되므로, 칩선택 신호 $\overline{CS3}$가 항상 데이터 버스의 종료보다 ½클럭 주기만큼 먼저 상승 에지를 나타내는 원리에는 변화가 없기 때문이다.

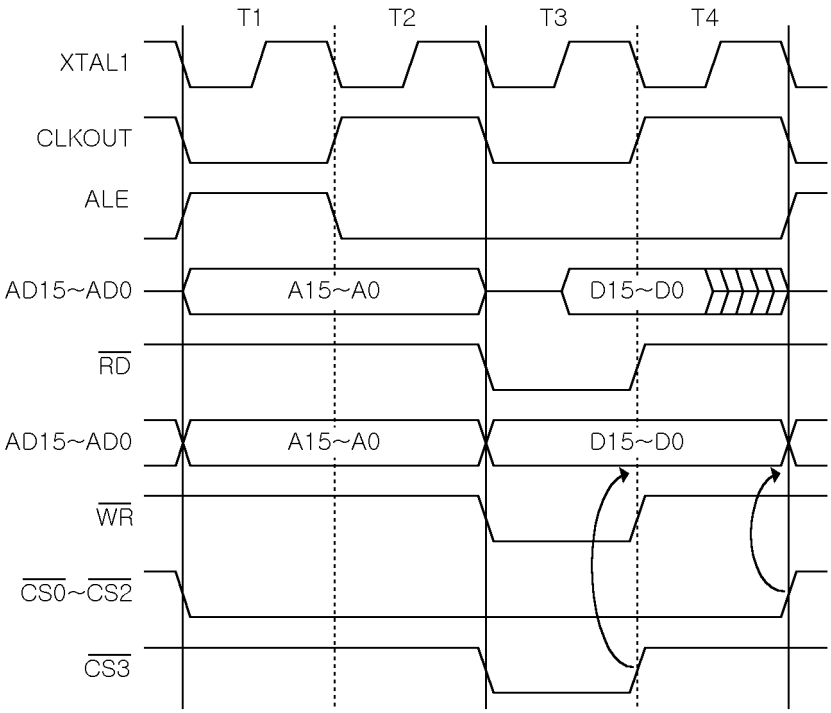
▣ 80C196KC 마이크로콘트롤러의 경우

80C196KC의 외부 액세스는 시스템 클럭을 기준을 할 때 기본적으로 4클럭 주기에 수행된다. 그러나, 이를 2분주한 CLKOUT 신호의 1주기를 스테이트(state)라고 하는데 이것으로 따지면 2스테이트가 된다.

이와같은 80C196KC의 외부 액세스 동작 타이밍을 그림으로 보이면 <그림 5>와 같다. 80C196KC는 특이하게 외부 핀수를 절약하기 위하여 어드레스 버스와 데이터 버스가 시분할 다중화(time-multiplexed)되어 동작하는 방식을 채택하고 있는데, 이 때문에 동작 사이클에서 첫번째 클럭 T1에는 여기에서 어드레스와 함께 이를 외부의 플립플롭에 래치하기 위한 신호 ALE가 출력된다. 그러나, 이 래치의 홀드 타임을 충분히 확보하기 위하여 어드레스 신호는 T2 클럭까지 연장되어 출력된다.

라이트 동작의 경우에는 T3~T4 동안에 데이터가 출력되면서 T3 동안만 $\overline{WR}$ 신호가 L상태로 출력됨으로써 데이터가 출력되는 파형의 중앙에 $\overline{WR}$ 신호의 상승 에지가 위치하도록 동작한다. 80C196KC의 동작 파형이 이렇게 만들어진 것은 물론 출력 래치에서 셋업 타임이나 홀드 타임이 충분히 확보되는 구조를 갖게 하기 위함일 것이다.

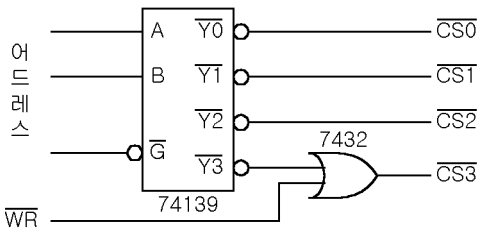
그러나, 리드 동작에서는 CPU가 정확히 $\overline{RD}$ 신호의 상승 에지에서 데이터를 읽어들이기 때문에 그 뒤에는 최소한의 홀드 타임을 확보하고 나면 더 이상의 데이터 입력은 무효하다는 것을 파형도에서 보여주고 있다.



<그림 5> 80C196KC의 출력 라이트 동작 타이밍

80C196KC의 라이트 동작이 이와 같이 수행되므로 병렬 출력래치 인터페이스 회로를 위한 칩선택 신호도 역시 이를 고려하여 만들어야 한다. 즉, <그림 5>에서 어드레스 신호만을 가지고 74139 어드레스 디코더로 디코딩하여 만들어진 칩선택 신호 $\overline{CS0}$ ~ $\overline{CS2}$는 홀드 타임을 충분히 확보할 수 없으므로 출력 래치에 사용할 수 없으며, 어드레스 디코더의 출력과 $\overline{WR}$ 신호를 부논리로 AND하여 만든 신호 $\overline{CS3}$는 항상 데이터 버스의 종료보다 1클럭 주기만큼 먼저 상승 에지를 나타내므로 래치에서 요구하는 충분한 홀드 타임을 갖게 되어 정확하고 안정된 래치 동작을 수행한다.

이러한 사항을 고려하면 80C196KC의 출력 라이트 동작을 위한 어드레스 디코더 회로는 <그림 6>과 같이 설계해야 한다. 여기서도 역시 RAM, ROM, 82C55A, D/A 컨버터 등은 $\overline{CS3}$ 신호가 아니라 $\overline{CS0}$ ~ $\overline{CS2}$의 신호를 칩선택 신호로 사용하여야 한다.

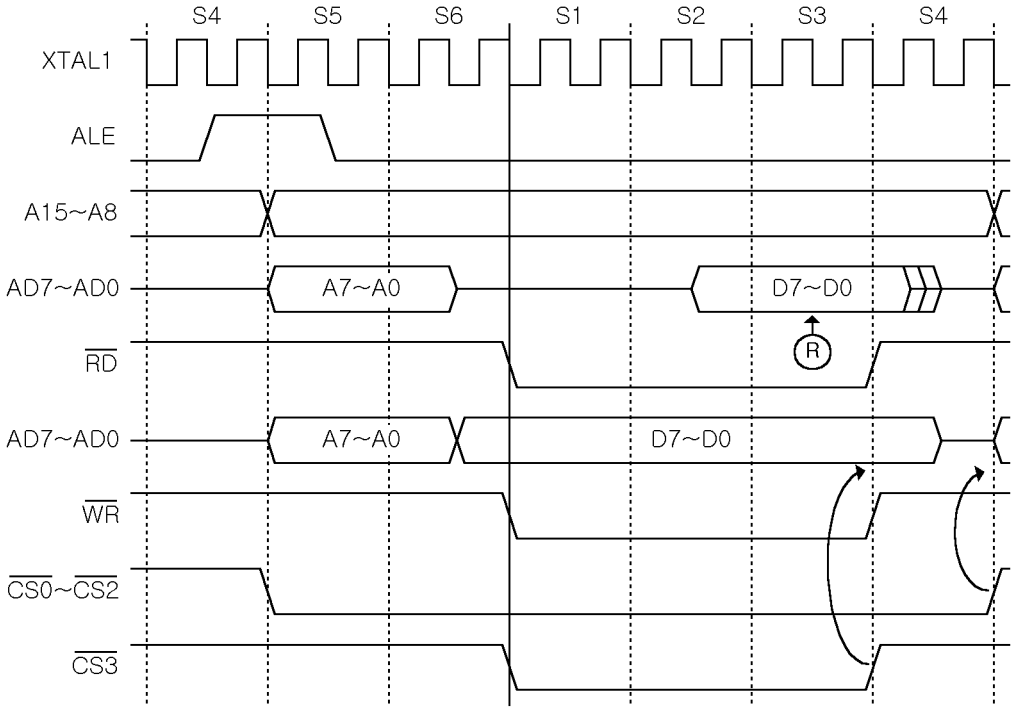


<그림 6> 80C196KC에서 올바른 칩선택 신호의 발생

또한, 이 회로의 경우에도 1개 이상의 웨이트 사이클을 갖는 느린 소자의 리드/라이트 동작에서는 <그림 5>의 T3 클럭 중앙 부분에 스테이트 단위로 필요한 웨이트 사이클이 추가되므로, 칩선택 신호 $\overline{CS3}$가 항상 데이터 버스의 종료보다 1클럭 주기만큼 먼저 상승 에지를 나타내는 원리에는 변화가 없으므로 아무 문제가 없이 잘 동작한다.

▣ 8051 마이크로콘트롤러의 경우

8051은 외부 메모리가 프로그램 메모리(ROM)와 데이터 메모리(RAM)로 구분되어 있는 것이 중요한 특징이며, 머신 사이클(machine cycle)은 항상 일정하게 12주기의 시스템 클럭 또는 S1~S6의 6스테이트로 구성된다. 그러나, 8051에서 대부분의 경우 프로그램 메모리에 대하여는 1개의 머신 사이클에 2차례의 명령 코드 페치(fetch)가 가능하여, 각각 S1~S3 스테이트에 1번 또는 S4~S6 스테이트에 1번 프로그램 메모리를 읽는 것이 가능하다.



<그림 7> 8051의 출력 라이트 동작 타이밍

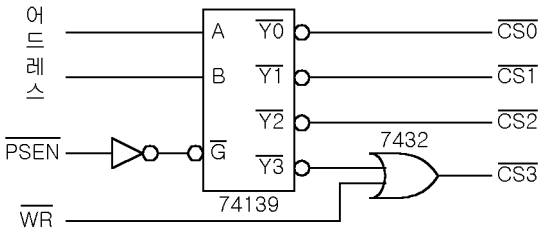
8051은 메모리맵 I/O(memory-mapped I/O) 방식을 사용하므로 모든 입출력 장치는 데이터 메모리 영역에 접속해야 한다. 데이터 메모리를 리드/라이트하는 것은 항상 MOVX 명령을 사용해서만 가능한데, 이것들은 기계어 코드가 모두 1바이트의 짧은 명령으로 되어 있다. 따라서, 이 명령의 첫번째 머신 사이클에서 S1~S3 스테이트는 OP 코드 페치가 수행되고, 나머지의 S4~S6 스테이트와 두번째 머신 사이클의 S1~S3 스테이트까지 6스테이트 동안

에 메모리 리드 또는 메모리 라이트 동작이 수행된다. 마지막의 S4~S6 스테이트는 아무 동작도 수행하지 않는 더미 스테이트들로 되어 전체적으로 2개의 머신 사이클 또는 24개의 클럭으로 이 명령의 실행이 종료된다.

이러한 메모리 리드 및 라이트 동작의 파형을 도시하면 <그림 7>과 같다. 8051도 역시 40핀 DIP형의 한계 때문에 외부 핀수를 절약하기 위하여 80C196KC에서처럼 어드레스 버스와 데이터 버스가 시분할 다중화되어 동작하는 방식을 사용한다. 이 때문에 S3의 후반부에서부터 S6의 전반부까지는 외부 래치에 어드레스값을 래치시켜 어드레스 버스를 분리하는 것이 기본적인 기능이며, 그 나머지 부분은 실제로 데이터를 리드 또는 라이트하는 부분이다. $\overline{RD}$ 및 $\overline{WR}$ 신호는 모두 2번째 머신 사이클의 S1~S3 스테이트 동안에 L상태로 되며, 라이트 동작의 경우에는 데이터 버스가 S6 스테이트의 후반부에 시작되어 S4 스테이트의 전반부까지 출력됨으로써 $\overline{WR}$ 신호의 상승 에지 이후에도 1클럭 주기의 홀드 타임이 보장되도록 되어 있다.

리드 동작도 대체로 데이터 입력 신호가 S2 스테이트의 후반부에서 S4 스테이트의 전반부에 걸쳐서 입력되는 것이 바람직한데, 다른 마이크로프로세서들이 일반적으로 $\overline{RD}$ 신호의 상승 에지에서 데이터를 읽어들이는데 비하여, 8051의 경우에는 $\overline{RD}$ 신호의 상승 에지가 발생하기 1클럭 앞에 해당하는 S3 스테이트의 중앙 부분(Ⓡ로 표시)에서 실제로 리드 동작이 수행되는 것으로 되어 있다.

8051의 데이터 메모리에 대한 라이트 동작이 이와 같이 수행되므로 병렬 출력래치 인터페이스 회로를 위한 칩선택 신호도 역시 이를 고려하여 만들어야 한다. 즉, <그림 7>에서 어드레스 신호와 데이터 메모리를 지정하기 위한 $\overline{PSEN}$ 신호만을 가지고 74139 어드레스 디코더로 디코딩하여 만들어진 칩선택 신호 $\overline{CS0}$ ~ $\overline{CS2}$의 상승 에지는 데이터 버스가 무효로 된 훨씬 이후에 나타나므로 출력 래치에는 사용할 수 없으며, 이 어드레스 디코더의 출력과 $\overline{WR}$ 신호를 부논리로 AND하여 만든 신호 $\overline{CS3}$는 항상 데이터 버스의 종료보다 최소한 1클럭 주기만큼 먼저 상승 에지를 나타내므로 래치에서 요구하는 충분한 홀드 타임을 갖게 되어 정확하고 안정된 래치 동작을 수행한다.



<그림 8> 8051에서 올바른 칩선택 신호의 발생

이러한 사항을 고려하면 8051의 데이터 메모리에 대한 출력 라이트 동작을 위한 어드레스 디코더 회로는 <그림 8>과 같이 설계해야 한다. 여기서도 역시 데이터 메모리 영역에 포함되는 RAM, 82C55A, D/A 컨버터 등은 $\overline{CS3}$ 신호가 아니라 $\overline{CS0}$ ~ $\overline{CS2}$의 신호를 칩선택 신호로 사용하여야 한다.

2. 입력단 3스테이트 버퍼 회로의 설계

CPU에서 외부로부터 병렬 입력 데이터를 읽어들이는 동작은 병렬 출력 데이터의 라이트 동작에 비하여 매우 단순하다. 이것은 대부분의 CPU가 $\overline{RD}$ 신호의 상승 에지에서 데이터 버스를 읽어들이는 기능을 수행하도록 설계되어 있기 때문이다.

병렬 출력 인터페이스 회로에서는 거의 대부분 74LS574와 같은 래치 회로를 사용하여 CPU가 출력 명령으로 외부 번지에 짧은 순간 동안에 라이트하는 데이터를 래치시켰다. 그러나, 병렬 입력 인터페이스 회로에서는 수많은 입력 장치중에서 칩선택 신호로 1개만을 선택하여 데이터 버스에 접속시키고, 그 짧은 수십 내지 수백 [ns]의 시간 동안에 CPU가 이를 읽어들일 수 있도록 해 주어야 한다. 이렇게 데이터 버스를 필요할 때만 CPU쪽으로 열어주는 기능에는 74LS244, 74LS245와 같은 3스테이트 버퍼가 사용된다.

따라서, 얼핏 생각하면 이러한 병렬 입력 인터페이스 회로에서도 출력 인터페이스의 경우처럼 칩선택 신호를 만들 때 어드레스 디코더의 출력에 $\overline{RD}$ 신호를 포함시켜야 할 것으로 생각하겠지만, 여기서는 이것이 오히려 바람직하지 않다. 즉, $\overline{RD}$ 신호가 끝나는 상승 에지에서 CPU가 데이터 버스를 읽어들이고 있는데 이 $\overline{RD}$ 신호와 동일한 타이밍의 칩선택 신호에 의하여 3스테이트 버퍼가 닫혀버리면 곧바로 데이터 버스가 끊어지므로 CPU가 이를 읽는데 필요한 홀드 타임이 충분히 확보되지 않을 수 있기 때문이다.

그러나, 실제로는 CPU는 아무 시간 지연이 없이 $\overline{RD}$ 신호의 상승 에지에서 데이터 버스를 읽어들이는데 비하여, 어드레스 디코더와 $\overline{RD}$ 신호가 결합하여 만들어지는 칩선택 신호는 이렇게 중간에 사용되는 디지털 소자들에 의한 시간 지연과 3스테이트 버퍼 자체의 동작 지연 시간까지를 고려하면 총 시간 지연값은 CPU에 필요한 최소한의 홀드 타임을 충분히 초과하게 된다. 하지만, 이렇게 하면 3스테이트 버퍼가 열리는 시간이 $\overline{RD}$ 신호의 폭만큼으로 줄어들어 자칫 CPU가 입력 장치를 읽어들이는 액세스 타임이 부족한 경우가 발생할 수도 있다.

이러한 이유로 입력 인터페이스 회로에서도 어드레스 디코더의 출력에 $\overline{RD}$ 신호를 결합한 칩선택 신호를 사용하더라도 대부분의 경우에는 아무 문제가 없지만, 특별히 논리적으로 꼭 그렇게 해야될 필요가 있지 않다면 그냥 어드레스 디코더의 출력만으로 칩선택 신호를 만들어 사용하는 것이 더 안전하다.

▣ TMS320C31 DSP의 경우

이 DSP의 경우에는 <그림 2>에서 보았던 것처럼 리드 동작이 무조건 H1 클럭으로 1클럭 주기에 수행되므로 어드레스 버스, R/$\overline{W}$, $\overline{STRB}$ 신호 등이 모두 1클럭 주기동안 출력된다. 따라서, 칩선택 신호는 여기에 $\overline{STRB}$ 신호를 포함시킬 것인지의 여부에 관계없이 동일하게 1클럭 주기의 신호밖에 만들어지지 않으며, 이것으로 3스테이트 버퍼를 제어하는 방법밖에는 다른 수단이 없다. 즉, <그림 3>에서 $\overline{CS0}$ ~ $\overline{CS2}$ 신호를 사용하든 $\overline{CS3}$ 신호를 사용하든 동작이 동일하다는 것이다.

이렇게 하더라도 앞에서 설명하였던 것처럼 칩선택 신호를 만드는 인터페이스 회로에서의 시간지연과 3스테이트 버퍼의 동작시간으로 충분히 리드 동작을 위한 홀드 타임이 확보된다. 다만, <그림 2>의 (a)에서 보인 것처럼 외부에서 입력할 데이터는 1사이클이 종료된 이후까지 약간 더 시간을 연장하여 공급하는 것이 바람직하다.

▣ 80C196KC 마이크로콘트롤러의 경우

80C196KC의 경우에도 <그림 5>를 고려하면 병렬 입력 인터페이스 회로의 칩선택 신호를 만들 때 $\overline{RD}$ 신호를 포함시키는 것에 관계없이 입력 동작은 문제없이 수행된다. 그러나, 이 그림에서 보듯이 4클럭의 입력 사이클 동작중에서 $\overline{RD}$ 신호가 인에이블되는 시간은 불과 1클럭 주기에 불과하므로, CPU의 클럭 주파수가 높고 입력 장치의 회로가 동작이 느릴 경우 자칫 액세스 타임이 부족해지는 사태가 초래될 수 있다.

이러한 점을 고려하면 특별한 사정이 없는한 80C196KC에서 병렬 입력 인터페이스 회로의 3스테이트 버퍼 제어 신호는 어드레스값만 디코딩하여 만든 칩선택 신호를 사용하는 것이 좋다. 즉, <그림 6>에서 $\overline{WR}$ 신호를 $\overline{RD}$ 신호로 대체하고, $\overline{CS3}$ 신호를 사용하는 것보다는 $\overline{CS0}$ ~ $\overline{CS2}$ 신호를 사용하는 것이 바람직하다는 것이다.

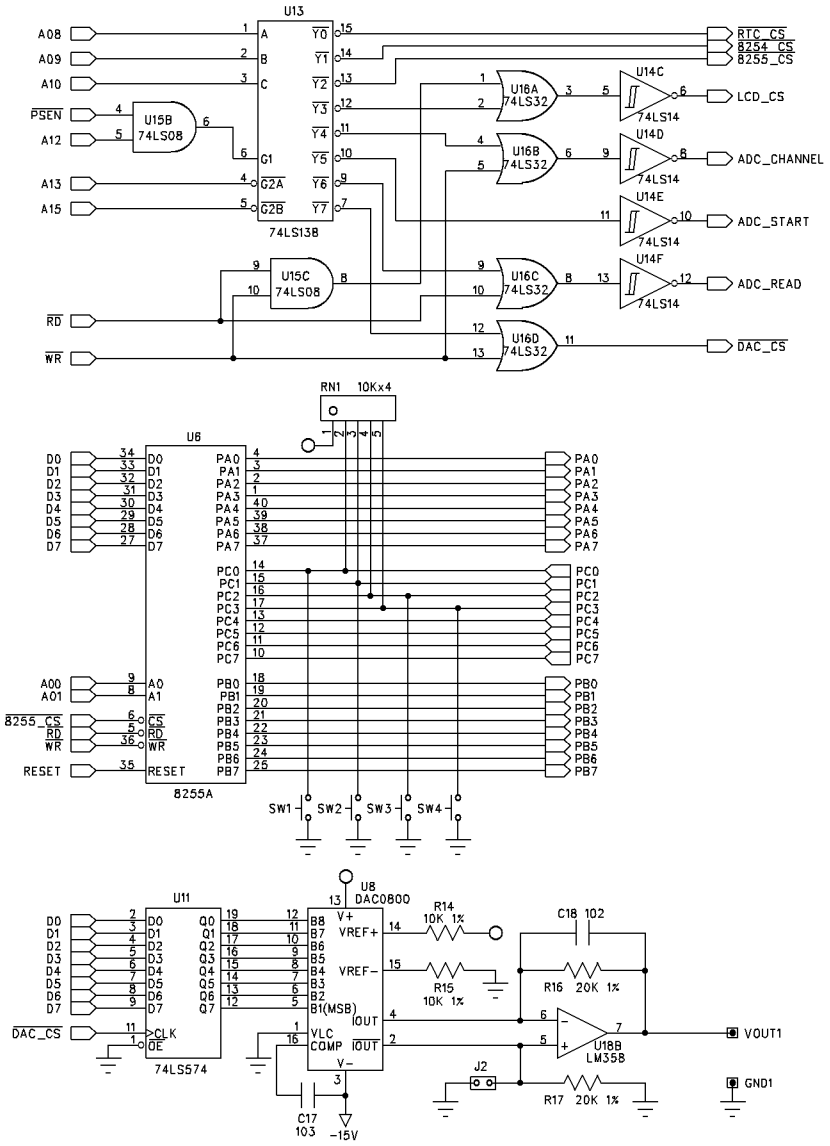
▣ 8051 마이크로콘트롤러의 경우

8051의 외부 데이터 메모리 액세스 동작도 앞의 <그림 7>에 보였던 것처럼 전체적으로는 80C196KC의 경우와 유사한 형태였다. 따라서, 병렬 입력 인터페이스 회로의 칩선택 신호를 만들 때 $\overline{RD}$ 신호를 포함시키는 것에 관계없이 입력 동작은 문제없이 수행된다.

그러나, 8051은 CPU가 특이하게도 실제로 데이터를 읽어들이는 시점이 $\overline{RD}$ 신호의 상승에지보다 1클럭 주기가 앞서도록 동작하며, $\overline{RD}$ 신호의 펄스폭도 3클럭 주기나 되므로 대부분의 경우 $\overline{RD}$ 신호를 포함한 칩선택 신호를 사용해도 별 문제가 없다.

그러나, 이 경우에도 아주 느린 입력 회로를 사용한다면 <그림 8>에서 $\overline{WR}$ 신호를 $\overline{RD}$ 신호로 대체하고, $\overline{CS3}$ 신호를 사용하는 것보다는 $\overline{CS0}$ ~ $\overline{CS2}$ 신호를 사용하는 것이 바람직하다.

마지막으로, <그림 9>에는 이상에서 설명한 병렬 입출력 인터페이스 회로의 설계 예를 보였다. 이것은 필자가 저술한 "어셈블리와 C언어로 익히는 8051 마스터"(Ohm사) 책에 부속되는 *OK-8051* 키트 회로도의 일부이다.



<그림 9> 8051에서 병렬 입출력 인터페이스의 설계 예 (*OK-8051* 키트)

【 참고문헌 】

1. 윤덕용, TMS320C31 마스터, Ohm사, 1988

2. 윤덕용, 어셈블리와 C언어로 익히는 80C196KC 마스터 (I), Ohm사, 2000

3. 윤덕용, 어셈블리와 C언어로 익히는 8051 마스터, Ohm사, 2001