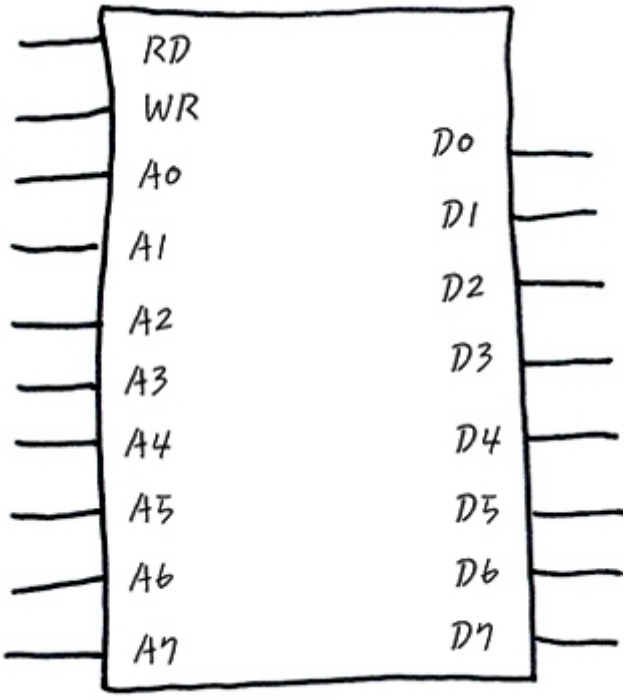
친절한 임베디드 시스템 개발자 되기 강좌 : RAM Memory의 물리적 동작

RAM Memory의 물리적 동작
Memory를 이해하면 System을 이해하는데 엄청 좋아요. Memory를 모르고는 System을 이해 한다고 말하기 좀 그렇죠. 모든 주변 Device들은 Memory와 똑같이 Control 가능하다고 보시면 되니까요. 더 정확하고 자세한 Memory의 Control에 관련해서는 Device Control에 System Architecture의 "Memory Device를 Control한다는 것"을 읽어 보시면 더 자세하게 나올테고요. 기본적인 Memory의 동작에 대해서 얘기해 보죠.

Memory는 이렇게 생겼어요.



Address Pin과 Data pin으로 구성되어 있고요, 나머지는 RD (Read), WR (Write) pin으로 구성되어 있어요. RD는 Memory로부터 Data를 읽을 때, WR은 Memory에 Data를 Write할 때 사용하는 pin이고요, A[0]~ A[7]은 Address pin, D[0]~D[7]은 Data pin이에요. Address는 개수가 8개니까, 2^8 가지 (각 pin 마다 0 아니면 1을 가질 수 있으니까요) Address를 나타낼 수 있으니까요. 0~ 2^8-1 주소만큼을 Memory가 처리할 수 있는 거지요. - 0x0~ 0xFF까지 에요, 256 Byte만큼의 크기네요 - 메모리 마다 처리할 수 는 양에 따라 Address pin의 개수가 틀려요. 1MB까지 처리 할 수 있는 녀석은 Address pin이 20개 있으면 되겠지요. (2^20이 1MB니까요) Data는 0~7까지 8bit를 처리할 수 있네요. 한번에 Data를 담을 수 있는 양이라고 보시면 되어요. 쓸 때도 1 Byte만큼, 읽을 때도 1 Byte 만큼을 한번에 처리할 수 있는 거죠. 그러니까 한번에 처리 가능한 양 만큼씩이니까 꼭 8 bit씩이라고 생각하시면 곤란해요.

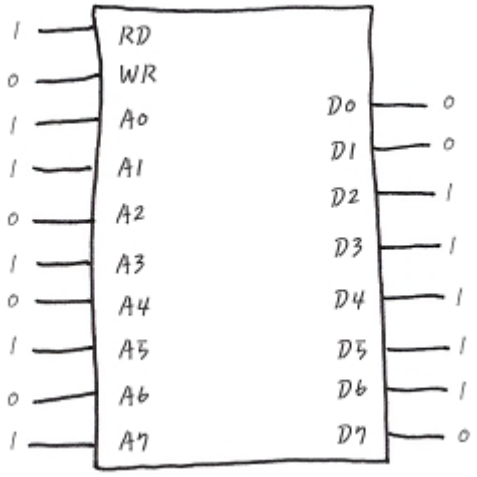
자, 그러면 이제 Memory가 어떻게 동작하는지 한번 볼까요! 0xAB (10101011)번지의 Data를 Write할 때의 Memory 상태는 이렇습니다요. WR에 1을 주고요, RD에는 0을 주는 거죠. Write할 꺼니까요.

그러면 0xAB번지를 Address pin에 주고서요, 쓰고 싶은 Data인 0x7C (1111100) 를 Data pin으로 주게 되면 0xAB번지에 0x7C가 써지는 거에요.

오호, 간단하죠. 그럼 반대로 읽을 때는 어떻게 할까요? 0x7C가 잘 써져 있는지 확인해 보시죠.

또 역시 간단해요. RD를 1로 주고요, WR을 0으로 주고요, Address에 0xAB를 주게 되면,

Data pin으로 0x7C가 튀어 나오게 되는 것이죠.



오호, 이런 거에요. Memory에는 이런 식으로 Data가 저장되고, 읽어드릴 수도 있는 거죠. 여기에서 RD나 WR은 다른 형식으로도 control될 수 있지만, (이건 뒤쪽에 더 자세히! 나올 꺼에요) Address와 Data에 관련된 건 아~ 주~ 똑같으니까, 이것만 알아둬도, 오호! Memory란 이런 거구나 하고 알 수 있을 것이라 생각하고 있어요.

Address Bus Width가 16 bit이면 최대로 나타낼 수 있는 Memory Size는 얼마 인가요? 라는 질문에 당당하게 대답하실 수 있는 부운~? 자자, 별거 아니지요~ 일단은 Address Bus Width가 16 bit이면 2^16 만큼 Address를 표현 할 수 있겠지요? 왜냐하면 1byte당 address를 1을 차지하거든요. 그러니까 2 byte면 0x2만큼의 Address가 필요하고요, 4byte면 0x4만큼의 Address가 필요한 거에요. 자자, 그러면 얼마 인가요? 2^16만큼의 Address를 표현 하는 거지요. 즉, 64KB이죠? 아하하~ 그러면 한번에 16 bit씩 Access하는 (즉, word가 16bit씩인 System) System에서는 얼마나 큰 Address 크기를 나타낼 수 있는 걸까요? 뭐, 간단하죠. 두배 되겠지요? 한번에 2 byte씩 Access하니까 굳이 1 byte씩 표현 할 이유가 없겠지요? 그러니까 64KB를 나타 낼 수 있는 Address line에서 그런 경우라면 128KB 인거죠. 이런 configuration에서는 dataline이 16개 나와 있을 거에요~ 그렇겠지요? 이거 잘 알아 두시면 엄청 편리하죠. - 이게 더 헷갈리게 하는 거 아닌가 모르겠어요. T.T