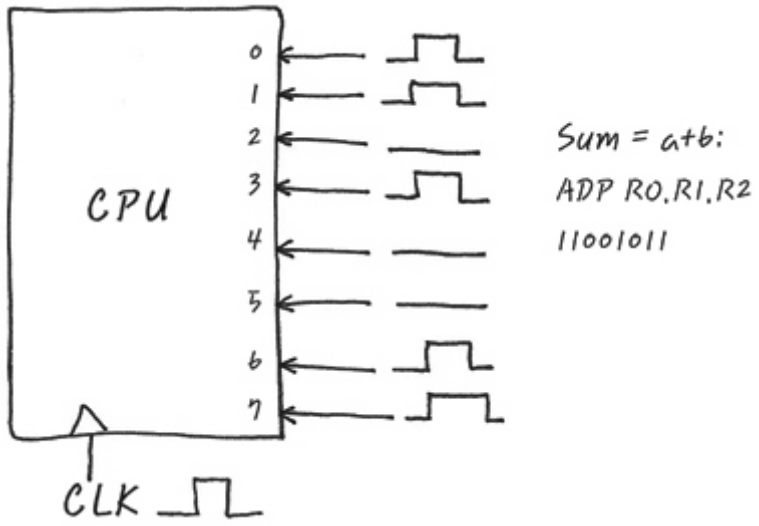
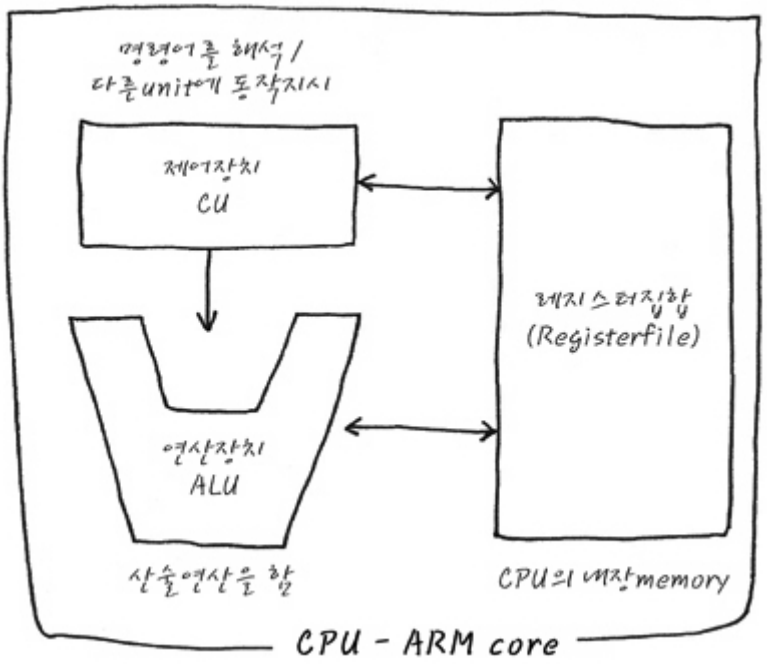
친절한 임베디드 시스템 개발자 되기 강좌 : 확장 to the CPU

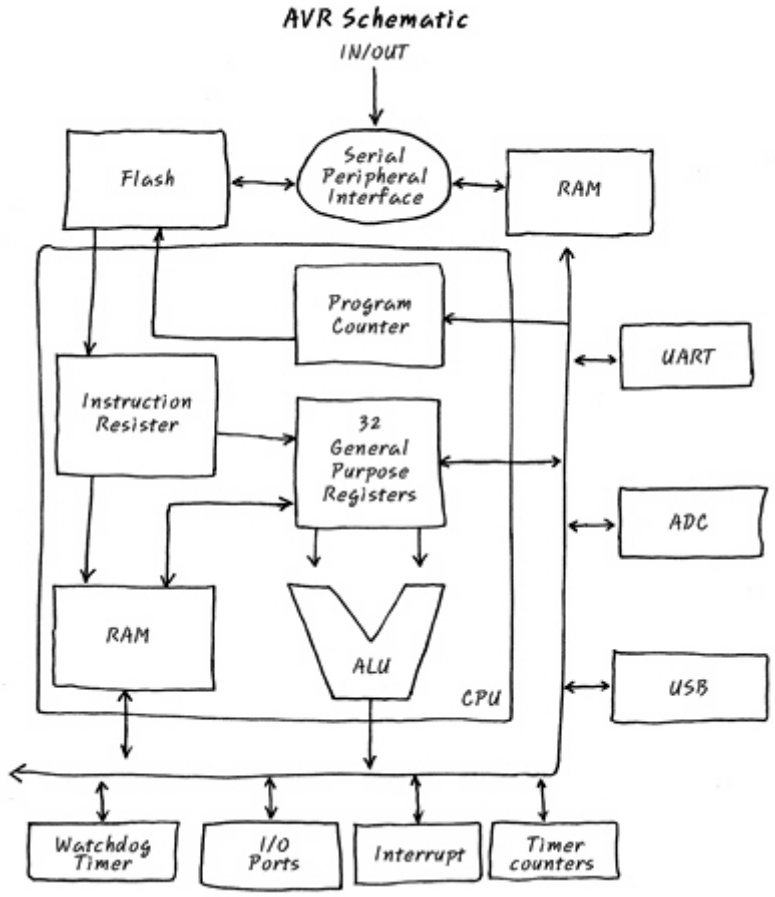
CPU라는 거, 이 녀석은 논리회로의 집합체 입니다. 논리회로 하나만 구성하면 한가지 일만 하면 되지만, 요 CPU라는 여러 가지 일을 해낼 수 있는 논리회로의 집대성체 - 인간으로 치면 두뇌에 해당한다고 할까요? 대신 학습능력은 "꽝" 이죠 - 로서 CPU에게 뭔가 정해진 일을 시키면, 그에 대한 일을 할 수 있도록 하는 역할을 합니다. 컴퓨터 프로그램의 명령어를 처리하기 위한 논리회로의 집합을 담고 있는 핵심 부품이라고 할 수 있겠네요. CPU에게 어떻게 일을 시키느냐 하면, CPU에 나와 있는 pin들에 약속된 신호를 주게 되면, 그에 대한 일을 CPU는 착실히 수행하게 됩니다. 결코 반항을 하거나, 비뚤어지지 져서 삐딱하게 굴지 않습니다.



그 일은 만일 8bit CPU에서 0~7번 pin에 약속 되어진 신호를 주게 되면 CPU는 그에 해당하는 일을 하게 되는 단순한 원리 입니다. 예를 들어 sum = a+b를 계산하고 싶으면 - 내부적으로는 Register R1에 a가, Register R2에 b가 들어 있고 그 결과를 Register R0에 저장하는 매커니즘으로 동작한다는 가정이라면.. - , 0~7번 pin에 위의 일을 하기 위해 약속 된 11001011이라고 전기적인 신호만 준다면 CPU는 내부의 Register R1과 R2에 들어 있던 값을 더해서 R0에 저장하는 일을 할 것입니다. CPU라는 건 상당히 단순한 논리회로의 집합체 인 것입니다. 단순한 일들이 모여 산을 이룬다고나 할 까요. 이런 말을 어렵게 하면, 이렇게도 표현합니다. 컴퓨터의 기계어 명령을 실행하기 위해서 수행되는 더욱 낮은 수준의 명령어. 이에 대해 원래 기계어 명령을 매크로 명령어 (macro instruction)라고 한다. 마이크로 명령어는 중앙 처리 장치(CPU) 내의 각종 게이트를 구동하는 비트 신호로 이루어지며 CPU 내의 제어 기억 장치에 저장되어 있다. 하나의 기계어 명령을 수행하기 위해서는 여러 개의 마이크로 명령이 수행되어야 하는데 CPU 내의 제어부가 마이크로 명령을 꺼내어 각 게이트에 신호를 준다.- http://kin.naver.com/openkr/entry.php?docid=23629 "정보 통신 용어 사전"이것도 맞고, 저것도 맞네요. 아하하. 이런 CPU도 내부는 또 다른 여러 가지의 장치들로 이루어져 있습니다. 예를 들어 제어장치 CU, 연산장치 ALU가 있으며 CPU내부의 저장 공간인 Register 집합체가 들어 있지요.



일단 CPU는 CU에서 명령어를 해석하여 다른 Unit에 동작을 지시하는 일을 하며 - 엄밀히 말하면 CU와 Decoder로 나눌 수 있는데, Decoder는 명령어를 읽어서 해석하는 일을 하며, 이에 대하여 CU는 각종 제어 신호를 발생하게 됩니다. 예를 들어 ALU에게 더하기를 하라는 신호를 발생시킨다던가, 또는 메모리에게 특정 주소를 Read할 수 있도록 신호를 발생시킨다든가 하는 여러 가지 Control signal들 입니다. ALU는 산술연산을 하게 됩니다. 또한 이런 산술연산이나, 제어에 관련한 결과 등을 임시 저장하는 Register file을 가지고 있습니다. 결국, 앞서 말했던 전기적인 신호 (명령)에 대해서 반응 하는 논리회로의 집합은 CU이고, 계산을 하는 논리회로의 집합은 ALU입니다. CU가 일차적으로 반응을 하면서 CPU내 외부의 모든 control을 하게 되는데, 그 control이라는 것도 다른 unit에서 볼 때는 약속된 전기적 신호를 말하는 것입니다. 이런 전기적 신호가 난무하는 세상이 CPU 내부의 세상이라고 생각하니, 울컥 하는 느낌입니다만, 어째 잘 정비된 도시의 도로 같기도 할 것 같은 느낌입니다. 예를 들어 이런 CPU에 Embedded system에 걸맞게 여러 가지 외부 기능이 더해진 AVR MCU는 다음과 같은 구조를 갖습니다. 잘 보면 CPU부분은 Program Counter (PC), 32개의 General Register들, 그리고, Memory로 부터 명령어 (약속된 일련의 전기신호)를 가져와서 일단 담아 놓는 Instruction Register (IR), ALU, 보너스로 내부 RAM정도를 CPU 부분에 가지고 있습니다. 지금 또 나왔는데, CPU라고 써 놓은 부분이 Core입니다. 나머지는 들러리, 즉 Peripheral block 들이라고 봐야겠지요. 뭐 어째 들러리라고 하니 푸대접하는 것 같아 좀 그렇습니다만 일단 이렇게라도 말해놓아야 Core가 주인공이 되는 거 아니겠어요?- 참고로, CPU -ARM Core와 1:1 matching해서 들여다 보면, CU가 안보이죠? 보통의 CPU block도에는 Data flow위주의 block도를 보여주니까, CU는 안 보이는 게 보통이죠. 자세히 보시면 Data에 관련되는 block만 보일 것이에요. -



CPU block 이외에 Flash, UART, I/O등 여러 가지 기능들의 논리회로들이 버스를 통해서 연결되어 한 개의 chip내에 implementation되어 있기도 합니다. 이런 식으로 CPU이외의 여러 가지 기능까지 덧붙여서 한 개의 chip으로 만든 것을 MCU (Micro Controller Unit) 이라고 부르기도 한답니다.

Memory로 부터 명령어 (약속된 일련의 전기신호)를 가져오는 일을 Pipe lining의 Fetch단계라고 부르는데, 이런 Pipe lining이라는 것도 살펴 봐야 하겠지요. 이런 시작 부분에서부터 골치 아픈 것을 다루면 지쳐서 잠들어버릴지 모르니, 그냥 언급만 하고 넘어가고, 본격적으로 다루는 건 다음으로 은근슬쩍 미루는 "재치"입니다.

소프트웨어의 정의는 : 전기적신호 11001011등의 전기적 신호는 외부메모리에서부터 흘러 들어오는 신호이며, 이런 신호가 바로 Software 입니다. 11001011가 바로 기계어라 불리 우는 약속된 패턴이며, 이런 패턴을 만들어 내기 위해서는 우리가 직접 이런 패턴을 만들어서 메모리에 넣어둘 수 있겠지만, C나 C++ 같은 인간이 이해하기 쉬운 언어로 인간이 Software를 만들면, Compiler를 이용하여 기계가 이해할 수 있는 기계어 패턴을 만들 수 있습니다. 기계어 패턴을 다른 말로는 Native Code라고 불러요. 바로 그런 인간언어를 기계가 이해할 수 있는 약속된 전기신호로 바꿔주는 과정이 compile입니다. 뒤에 나오는 컴파일에 대한 단상을 보시면 다시 한번 자세히 제 의견을 피력했으니 잘 봐주시면 감사.

by 히언 | 2009/06/01 21:35 | 하드웨어꼴라쥬 | 트랙백 | 핑백(2) | 덧글(2)