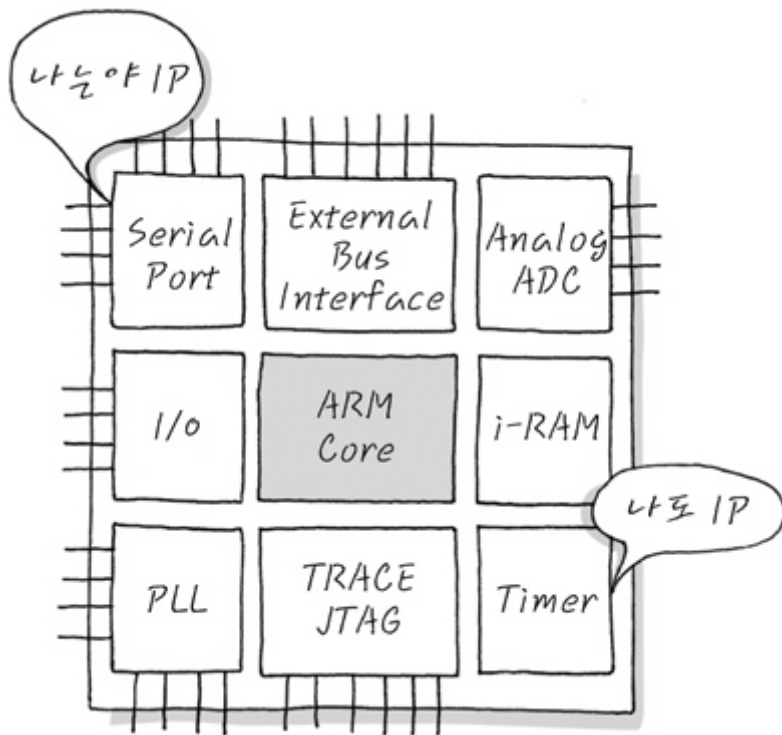
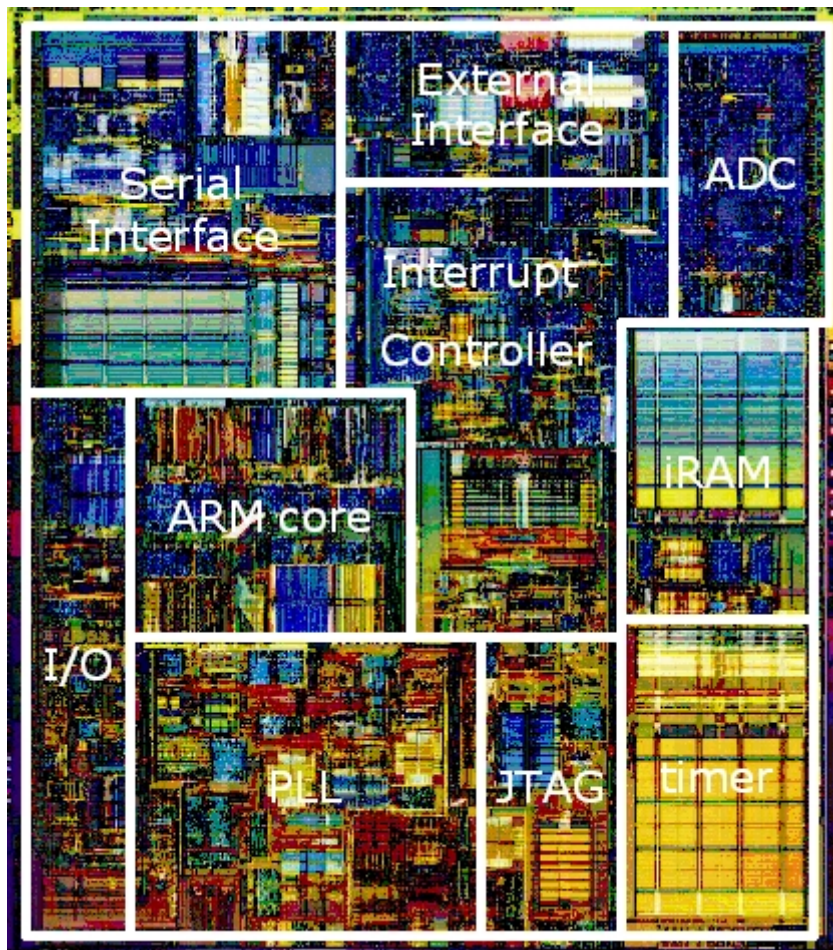


# 친절한 임베디드 시스템 개발자 되기 강좌 : ARM SoC (System On Chip)

ARM이라는 회사는 참으로 대단한 회사 입니다. 경영전략이 참으로 뛰어난 회사라는 의미입니다. 제가 처음 연구소에 다니기 시작했을 때, ARM core라는 말을 많이 들었는데, ARM이면 ARM이지, core는 또 뭐냐 - 라는 생각은 저만 하는 건가요? - 고 의문을 품었던 적이 있습니다. ARM core라는 단어 자체가 바로 ARM이라는 회사의 경영전략이 뛰어나다는 것을 의미합니다. ARM그렇고말고 - 꼭 이 말장난은 해보고 싶었습니다. ARM그렇고 말고 - ARM이라는 회사는 실제 Processor를 만들어서 파는 회사가 아닙니다. 그럼 무얼 팔아 먹고 사느냐, 바로 CPU Architecture를 파는 회사죠. ARM은 ARM이라고 이름 붙인, CPU architecture를 Design하고, 그것을 open한 다음, 그 architecture를 가져다 쓰는 회사로부터 로열티를 받아 그것으로 Business를 하는 회사입니다. 이런걸 일컬어, ARM license를 가져다 쓴다고 들 합니다. 보통은 MCU (Micro controller unit) 또는 MPU (Micro processor unit)를 만들어 Business를 하는 회사들이 이런 ARM license를 사서, 자기네 들의 Micro controller에 CPU 자리에 하는 ARM core를 장착하여 제품들을 만들어 냅니다. 그런 회사들은 Qualcomm, Samsung, Nokia, Intel, TI등이 그러하며, 이 회사들은 ARM core design의 license를 사서 자신들의 MCU, MPU에 맞게 다시 Design하기도 합니다. 그 예로는 Intel의 SA시리즈들이 그런데, 이 시리즈는 Intel이 ARM 9 core design을 변형하여, Strong ARM이라는 core 형태를 만들어 내어 이것을 장착하여 생산한 MPU입니다. 이런 chip들은 결국에 모두 같은 ARM core를 쓰다 보니까, ARM assembly를 모두 공용으로 사용하며, ARM Assembly를 이용해서 MCU를 control가능하게 합니다. 그 중 적당한 수준의 MCU는 Qualcomm MSM시리즈가 있는데, 이 Chip에는 심지어, Modem 기능이 ASIC되어 embedded되어 있습니다. ARM이 이런 core design만 팔다 보니, core에 여러 가지 기능을 덧붙여서 한 개의 chip으로만드는 기술이 스포트라이트를 받게 되었는데 이런 구조를 SoC라고 부르기도 합니다. 한마디로 System on Chip이라고 해서, ARM core를 바탕으로 웬만한 Device들을 block화 해서 한 chip에 집어 넣겠다는 음흉한 속셈인 것입니다. 오, 편리하지요. 이런 concept을 이용해서 여러 chip을 만들다 보니까, SoC 측면에서 각각의 기능들을 가진 chip을 각각의 block으로 만들면 재사용 가능하다는 것을 깨달아 버렸습니다. 모듈화 한다고 봐야죠. Clever! 이런 각각의 기능을 가진 SoC내부의 block들을 IP (Intellectual Property)라고 명명하기에 이르렀습니다. 이런 모듈들은 HDL이나, Verilog 같은 Design tool을 이용해서 모듈을 설계하고, ASIC Library를 이용해서 최종 netlist 까지 만들어 내면 하나의 모듈이 되고, 이런 모듈 하나를 IP라고 부른다고 보시면 됩니다. 시중에 많이 떠다니는 IP module들은 USB, UART, DMA 등등이고, 만약에 IP designer로서 제대로 된 IP하나만 가지고 있으면, 그 IP를 쓰고자 하는 회사들이 있으면 돈을 주고 IP를 사다가 쓰기도 합니다. 돈이 되는 일입니다요.



이런 것들이 모여 모여 각자의 기능으로 묶인 block들이 모여 한 개의 MCU라는 걸 이루게 됩니다. 웬만한 기능들은 이 chip안에 모두 가지고 있죠. Serial port라던가, I/O, PLL 등이 들어 있습니다.



뭐 좀 재미나게 그리면 이 정도가 되지 않을까 싶습니다. 그냥 보기에는 하나도 안 재밌고 어지럽기만 하네요. 이런 식으로 IP위주로 SoC를 하다 보니까, 이런 MPU내부에서도 서로 연결을 어떻게 해야 하느냐에 대한 고민이 생깁니다. 그냥 block만 끼워 넣어서 끝나면 좋겠는데, 어차피 이런 IP들끼리도 서로 interaction이 필요한 것 아니겠습니까. 이런 IP구조의 SoC도 역시 intercommunication을 어떻게 하느냐에 대한 고민은 또 ARM社에서 풀어 주었습니다. 이런 SoC가 아니더라도 보통 Digital Chip들은 내부에서 Bus를 통해서 Data가

흘러 다니는데 그런 의미에서 ARM社は 내부 IP들 끼리 Bus를 통해서 서로 잘 통신할 수 있도록 Bus Protocol을 제안했는데, 그게 바로 AMBA입니다.- 왜 이 AMBA protocol이 각광을 받느냐 하면, ARM社가 ARM core와 어떻게 통신하면 가장 효율적인가에 대한 연구를 많이 했으며, 게다가 이 Protocol은 Free입니다. 정말 경제논리에 의해 공짜 라는 게 큰 감동으로 다가올게 뻔하지 않겠습니까. - 결국 이런 ARM SoC chip의 내부에서는 AMBA Protocol을 통해서 서로 intercommunication을 합니다.



MCU는 유재석? 이라는 질문을 받은 적이 있습니다 .5초 후에 폭소해 버리긴 했지만.



MCU와 MPU의 구분에 너무 혼동할 필요는 없습니다. MCU는 Controllor의 성격이 강한 그러니까, CPU이외의 것들이 모두 들어 있어 Micro Computer처럼 혼자 동작할 수 있는 것을 말하고, MPU는 CPU의 성격이 강하여 외부 Device들이 있어야 동작 가능한 것을 얘기합니다, 그러나, 실제로는 많은 것들이 들어 있기도 하고, Memory같은 것들은 External Device로 나와 있거나 하는 등의 적당히 집적화 되어 있으면서도 동작하기 위해서 외부 Device도 필요한 수준의 Chip들도 있습니다. 뭐 어중간 하다고나 할까요? 이런 chip들의 장점은 적당히 집적화 되어 있어서 싸기도 하고, 외부에서 동작을 위해 같이 implementation해줘야 하는 device들에 대해서는 적당히 융통성을 부릴 수 있습니다. 더 싼 vendor것이 있다면 그런걸 써도 되니까요. 그러니까, 너무 집착할 필요는 없다니까요. 제 의견으로는 MCU냐, MPU냐를 구분하는 것처럼 시간 낭비도 없습니다. 저는 MCU든 MPU든 모두 CPU의 기능을 가지고 있으니까, 이제부터 구분 없이 MCU라고 부를 생각입니다만