Milestone Intro

by kkh

#1

아 뭐 텀프로젝트 하는데 도움이 되는 뭔가를 만들겠다고 생각하고 시작하려했는데 뭐부터 할지 막막하네요. 생각나는대로 써보겠습니다.

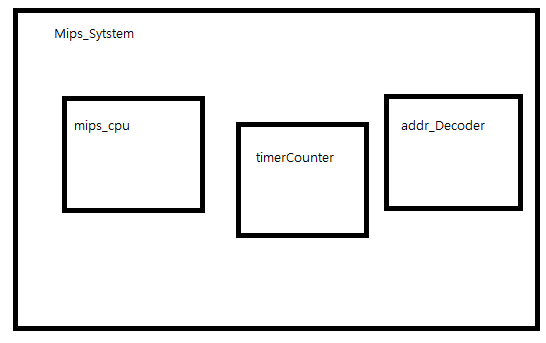
#2 이게 좀 길겁니다.. 중요해요 ★

마일스톤은 컴퓨터 구조 과목의 장기 프로젝트고 이 프로젝트를 마무리 하는 순간 꽤 엄청난 자부심과 뭔가 컴퓨터에 대해 진짜 많이 알게 된 것 같은 자신감이 생기는 묘한 숙제입니다. 자기 손으로 모든 것을 다 해낸다면 말이죠.. 무튼 그만큼 고통스러운데 이게 왜 그러냐면 투자한 시간이 결과를 전혀 보장 안해준다는 겁니다. 밤새서 완성해서 돌렸는데 안 되는일이 너무너무너무너무 x 100 비일비재 해서요. 그래서 이것을 시작하실 때 꼭 갖추셔야할 소양이.. **꼼꼼함**이랑 **인내심**이 꼭 있으셔야합니다..

무튼 시작하자면 여러분은 지난 학기 Verilog-HDL을 배우셨습니다. HDL이란 하드웨어 디자인 렝귀지. quartus ~~~.v 에 쓰셨던 그 많은 코딩들은 c언어로 소수들만 출력하는 프로그램 같은 소프트웨어가 아니고 하드웨어, 그니깐 기계를 만드는 겁니다. 이걸 무슨 c프로그램 짜는 것처럼 이 사실을 숙지하지 않고 시작하는 친구들이 많은데 그게 아니라 진짜 여러분 컴퓨터에 있는 cpu의 내부나 그 주변 메모리나 뭐 그런 것들을 언어로서 표현하는 것을 hdl이라고 하는거죠. 무튼 디테일한 문법 같은 것은 프로젝트 시작하시면서 봐도 상관없으나 정말 내 머리속의 지우개가 들어서 아무것도 생각이 안나시는 분은, milestone2할 떄부턴 컴논설 피피티 한번 보고 들어가시면 됩니다.

Tool들부터 간단히 설명하죠.

1. Quartus ★★★★★  
   - verilog로 mips cpu 및 mips system 디자인
2. Modelsim ★★★★★  
   - 시뮬레이션 하는 툴.
3. Eclipse ★  
   진짜 거의 필요도 없는데 프로젝트 시작하기 전 컴파일 한번 하는 용도. 컴퓨터가 먹통이라 크로스컴파일러 안 깔린 사람도 상관없음. 친구보고 컴파일 하라고 한 뒤 파일 받아도 무방

=> mips\_design 파일을 받아 quartus 파일을 실행하면 왼쪽 상단 project navigator에 꽤 여러개의 verilog 파일을 보실 수 있을 겁니다. 써있는걸 보면 Mips\_System 안에 addr\_decode, ram2port\_...., mips\_cpu, timerCounter 뭐 등등 있네요. 촉이 좀 날카로운 분이시라면 이 프로젝트는 하나의 시스템임을 알 수 있을겁니다. 하나의 모자란 컴퓨터를 디자인 해 놓은 것으로 보셔도 될 것 같네요. 그 중 여러분이 손을 대서 고치실 곳은 mips\_cpu입니다.

실제 mips\_cpu는 꽤 여러 가지의 instruction을 실행하죠. add sub부터 시작해서 beq, jump..등등 까지 배운것들 말이에요. 그런데 mips\_cpu.v를 열어서 주석 같은 것을 유심히 보시면, 지금 초기상태는 정말 극히 일부 dataprocessing instruction과 memorry access instruction만 처리가 가능하게 되어있습니다. beq, bne, jump, 뭐 이런것들은 하나도 없는 상태. 그럼 텀프로젝트는??이런 다양한 instruction들을 실행 할 수 있도록 코드를 고치고, 많~~이 더하고 하는 작업입니다. 그리고 저런 instruction이 실행 가능하게 여러분이 디자인을 다 하시고 나면, single cycle cpu에서 pipeline cpu로(그 laundry 그림 아시죠????) 업그레이드를 해주고, pipeline에서 발생하는 문제점들을 해결 할 수 있도록 디자인 하고 마지막 alu도 수정해서 여러분이 아는 모든 instruction은 돌릴 수 있는 pipeline cpu를 만드는게 최종 목표가 되시겠습니다.

그런데 곰곰이 한번 생각해 보`세요. 지금 이 코드들은 하드웨어 디자인만 해놓았지 메모리에 뭐를 넣어주고 이런 작업은 하지 않는 것으로 보입니다. cpu는 program counter가 메모리의 instruction을 가리키고, 4씩 더해지면서 실행을 하게 되는건데 메모리가 깡통이면 cpu는 그냥 아무것도 하지않는 애가 되지 않을까요? 우리가 이 cpu가 정상적으로 돌아가는 모습을 보기 위해서는 그럼 **메모리에 뭔가를 넣어주고!, 원하는 대로 돌아가는지 확인!을** 해줘야 합니다.

그럼 첫 번째 **메모리에 뭔가를 넣어주는 것**은 바로 eclipse를 이용해서 만듭니다. c코딩을 한 것을 cross compile해서 mips 기계어, 어셈블리어를 만드는 작업을 이미 했었죠. 그것과 똑같은데 그 컴파일 과정을 통해서 insts\_data.mif 라는 파일이 생성 됩니다. 이 파일은 그러니까 메모리 어디에 무슨 instruction을 넣어라~ 라는 파일인데요, 이 결과물을 mips\_design 폴더 어디에 잘 넣어주면 저 instruction들이 mips\_system 메모리 안에 예쁘게 들어갑니다.

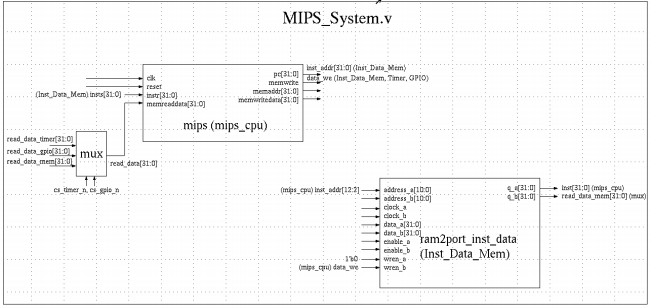
그럼 두 번째 **원하는 대로 돌아가는지 확인**을 해줘야겠죠. 뭐 단순하게 보드위에서 실행을 해보면 됩니다. 그럼 교수님이 의도한 LED나 불빛이 번쩍이거나 7segment로 숫자가 표시되거나 그런게 뜰테니깐요. 혹은 model\_sim에서 시뮬레이션 하는 겁니다. 이걸 할 줄 알아야 여러분이 앞으로 500줄 이상이 되는 코드에서 뭘 잘 못했는지 디버깅이 가능하기 떄문에 진짜 코딩하는 것보다 model sim에서 웨이브폼 읽는 법이 더 중요합니다. 컴논설에서 읽던거랑은 아예 체급이 달라요. 변수가 뭐 거의 몇 백 개씩 있으니깐, 원하는 타이밍에 어디에 무슨 값이 들어가는지 잘 볼 줄 알아야 합니다. 실제로 milestone ta께 검사 받을 때 뿐만 아니라 교수님과 인터뷰 할 때에도 modelsim으로 설명하라고 하시거든요. 한 학기에 이걸 똑바로 할 줄 알도록 연구한 사람은 1/3 채 안되는 것 같습니다. 전부 할 줄 아는 사람보고 디버깅 제발 해달라고 하죠. 여러분은 서로 다 할 줄 알아서 정보 공유하고 같이 잘 해쳐나가시길 바랍니다. 나중에 디버깅 하는 법 알려드리는 전공도우미 시간도 한번 가질 겁니다.

# 3

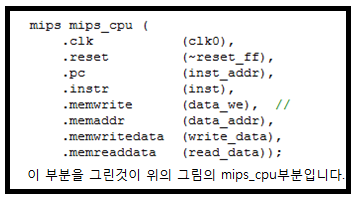
자 그럼 이제 진짜 여러분이 필요하실 milestone 1,2는 뭘 하는걸까요?

milestone1 은 한 줄로 말할 수가 있는데, c코드를 eclipse로 크로스 컴파일 하여 생성된 mif파일을 mis\_design폴더에 넣고 실행해보는겁니다. 근데 주어진 c코드가 하나의 7segment에 숫자 하나만 출력을 하게 되어있는데,,,,,,,,여러분의 학번을 출력하도록 그럼 수정을 해야겠죠?? 모두 5분안에 끝내실수 있을거라 생각합니다.

milestone2는 그림을 그리는 것입니다. 말 그대로 진짜 그림이죠. mips\_cpu.v 와 Mips\_system.v 의 구조를 스케치하는거에요.



뭐 이런식으로 말이죠. 이를 하는데에는 module의 input output을 잘 보시고 이게 어디랑 연결이 되는지를 포인트로 보시면 되겠죠. 가령 위의 그림으로 mux output read\_data[31:0]이거랑 cpu input memreaddate[31:0]하고 이어주는 wire 가 있을겁니다 그런것들 유심히 보시면서 어디랑 이어지는지 이 모듈에는 뭐가들어오고 뭐가나가는지 유심히 보시면 되겠습니다.

그런데 지금 그림은 좀 부품을 표시하고 연결되는 부분의 이름을 적어놓았는데 저는 가능하면 선까지 완벽하게 이어놓으셨으면 좋겠습니다. 잇고 나서는 wire 같은거 변수명도 다 써주시면 금상첨화구요. 왜냐면 이걸 똑바로 완벽하게 하고 나면 진짜 큰 그림이 그려져서 남들보다 나중에 훨~씬 더 수월하게 마일스톤 진행 할 수 있거든요. 마일스톤 배점으론 5점이지만 무시하시면서 하시면 나중에 정~말 골 아플 수도 있습니다. 그리신 다음엔 꼭!!!!!!!! 복사를 하시던지, 파일로 저장을 해두시던지 하길 강추 합니다.

#5

아 피곤하네요 이젠 놀러 나가야겠습니다. 모두 시험 끝난거 축하드리고요 혹시 남은시험이 있다면 잘 보세요!