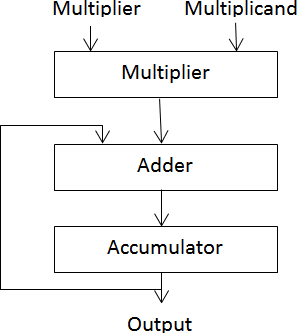
**디지털 논리회로 PA report**

201911039 김태연

A. Design Flow



Multiply-accumulator unit의 모식도는 위 그림을 참고했다. 두 개의 input, in과 w를 입력받아 multiply하도록 (\*) 연산자를 사용하였다. 기본적으로 multiply된 결과값을 accumulator의 output에 누적하는 모델을 구현하고자 하였다. 이 때, input이 준비되었을 때(DATA\_STROBE) multiply작업이 수행되고, positive clock edge에서 각 단위들의 output이 다음으로 넘어가기 위한 register에 업데이트되는 것으로 구성했다. 이를 위해서 각종 flag들과 state parameter를 활용하였다.

clock flow는 테스트를 통해서 확인하고자 했다. Clock의 주파수를 input과 weight가 입력되는 주파수의 절반으로 설정하여, 들어간 인풋이 업데이트되지 않는 상황을 피하고자 했다. Input의 준비 상태에 대한 flag(data\_strobe)가 작동하는지 확인하기 위해, data strobe 상태를 input, weight에 대해 (0,0),(1,0),(0,1),(1,1)의 입력을 주었다.

테스트에 대한 디자인은 상기와 같이 하였으나, 테스트벤치 및 모듈의 시뮬레이션은 실패하였다. 따라서 코드의 기능을 검증하지 못했음을 미리 밝힌다.

B. MAC sub-module and MAC Operator

MAC sub-module은 mac.v 파일에 구현되어 있고, MAC operator는 testbench에 구현하였다. MAC module의 parameter로 K를 선언하여, K=4, 8, 32에 대해 MAC operator을 따로 생성하고 시뮬레이션하는 코드를 구성하였다. 다음은 MAC module의 작동 방식에 대해 정리한 목차이다.

1. 클럭의 상승에지마다 state를 업데이트한다. 만약 reset signal이 들어와 있는 상태라면 IDLE 상태, 아니라면 동작 혹은 대기 중인 상태로 취급한다.
2. 다음 상태에 대한 결정은 상시 이루어진다. 만일 들어와있는 input 에 대한 data\_strobe가 low라면 데이터가 준비되지 않은 상태이므로 wait 상태에 돌입한다. 만일 모두 준비된 상태라면 MAC이 동작하는 상태로 주어진다.
3. BUSY\_FLAG 또한 상시로 업데이트된다. 이 플래그는 상태가 MAC, 즉 MAC operation이 동작하고 있을 때에만 BUSY이고 아닌 경우 NOT-Busy이다.
4. Feeding과 출력 업데이트는 순차적 회로의 clock에 따라 결정된다. 만일 clock이 positive이면 input이 레지스터로 들어와 계산 가능한 상태가 된다.
5. Counter는 accumulator에서 더해진 multiplied output의 개수를 세는 역할이다.
6. Reg\_mult의 결과물은 counter가 범위 내에 있을 시 reg\_out으로 전달되어 기다리고 있게 된다. 이 값은 K개의 input set에 대한 accumulation이 끝나고, clock signal이 positive edge일 때 mac\_out 단자로 전달된다.

이 동작들의 결과는 mac.vcd에 저장되게끔 했다.

본 코드는 다음 github의 legacy code를 참고하여 작성되었다.

https://github.com/roo16kie/MAC\_Verilog/blob/master/mac\_tb.v