

## Homework #2

### <8-bit Addition/Subtraction Module>

#### 과제 학습목표

---

- 선행과목 내용의 복습 (디지털논리회로 실습)
  - ✓ 기본적인 verilog syntax
  - ✓ adder, subtractor implementation
- module interface 가 주어졌을 때 module implementation 할 수 있는 능력을 함양
  - ✓ Interface를 만족하지 않는 모듈은 다른 모듈과 연동할 수 없음

#### Tips

---

- 아래 용어 및 개념이 생소할 경우 구글링하시길 권장합니다
  - ✓ 2's complement (Two's complement)
  - ✓ MSB (Most Significant Bit), LSB (Least Significant Bit), MSB first, LSB first
- Verilog 관련 자료는 YSCEC에 업로드된 자료 뿐 아니라 다른 자료를 많이 찾아보길 권장합니다
- 과제물 제출 전 최소한 sample input 으로 본인이 작성한 모듈을 테스트해보길 권장합니다
- 과제물 제출 전 마지막으로 다시 파일들을 열어보길 권장합니다

## Module Interface

---

- 모듈의 이름 및 입출력 포트는 반드시 다음과 같이 작성할 것 (대소문자 유의)

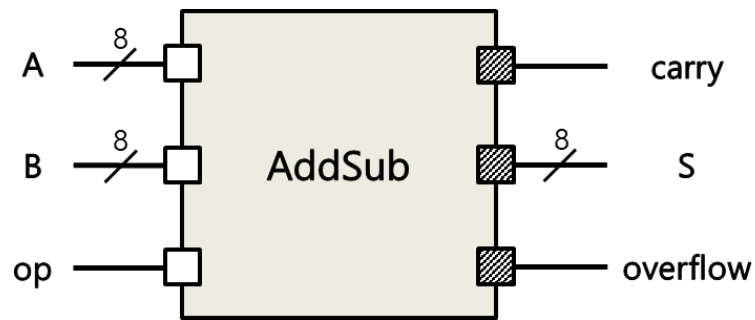


Figure 1

- module `AddSub`(A, B, op, carry, S, overflow);
  - ✓ `A`, `B` : 8-bit input, signed number (in 2's complement, MSB first)
  - ✓ `op` : 1-bit input, 0 (Addition) / 1 (Subtraction)
    - ◆ `op=0` (addition) :  $A + B$  값을 구하는 연산을 수행
    - ◆ `op=1` (subtraction) :  $A - B$  값을 구하는 연산을 수행
  - ✓ `carry` : 1-bit output
  - ✓ `S` : 8-bit output, signed number (in 2's complement, MSB first)
  - ✓ `overflow` : 1-bit output
    - ◆ `overflow=0` : No overflow.  $S = A + B$  or  $S = A - B$
    - ◆ `overflow=1` : Overflow.  $S \neq A + B$  and  $S \neq A - B$
  - ✓ 모든 입출력 포트는 MSB first
    - ◆ (ex) input `[7:0]` A;

## Sample Input & Output

Inputs			Output			Decimal
A	B	op	carry	S	overflow	
00001100	00001100	0	0	00011000	0	12+12 = 24
00001100	00001100	1	1	00000000	0	12-12 = 0
00001101	00011001	1	0	11110100	0	13-25 = (-12)
01100100	00110010	0	0	10010110	1	100+50 = 150
10110000	00111100	1	1	01110100	1	(-80)-60 = (-140)

In 2's complement, an 8-bit number is in the range of  $(-128) \sim 127$ .

Therefore,  $-128 \leq A \leq 127$  and  $-128 \leq B \leq 127$ .

Let  $\mathcal{R}$  be a 16-bit number which is the result of addition/subtraction generated by the module.

Then  $\mathcal{R} = A + B$  or  $\mathcal{R} = A - B$ .

$$-256 \leq A + B \leq 254, -255 \leq A - B \leq 255 \Rightarrow -256 \leq \mathcal{R} \leq 255$$

Note that  $S = (\text{last 8 bits of } \mathcal{R})$  always holds.

In conclusion, there are only three cases :

✓ (case 1)  $-128 \leq \mathcal{R} \leq 127$

overflow = 0, carry = 0 or 1

In this case, we only need 8 bits in order to represent the result of addition/subtraction.

✓ (case 2)  $127 < \mathcal{R} \leq 255$

overflow = 1, carry = 0

In this case, there is an overflow, thus the result of addition/subtraction can not be expressed with an 8-bit number. Note that the first 8 bits of  $\mathcal{R}$  is 00000000.

✓ (case 3)  $-256 \leq \mathcal{R} < -128$

overflow = 1, carry = 1

In this case, there is an overflow, thus the result of addition/subtraction can not be expressed with an 8-bit number. Note that the first 8 bits of  $\mathcal{R}$  is 11111111.

## 과제 제출 형식

---

- 과제 제출 형식을 준수하지 않을 시 감점
- 주어진 module interface 를 만족하는 구현은 모두 정답으로 인정. module interface 에 명시된 모듈명 등을 맞추지 않아 채점 불가시 감점
- 제출 기한 : 2018년 5월 17일 23:59:59
  - ✓ 5월 17일 이후 ~ 5월 24일 이전 제출 시 일괄적으로 만점의 30% 감점
  - ✓ [5월 24일 실습수업에서 해설해줄 예정이므로](#), 5월 24일 이후 제출 시 0점
- 제출물 : verilog 소스 파일 및 보고서 파일을 한 폴더에 넣고 압축한 [.zip 파일](#)
- "Lab1\_학번\_이름"으로 폴더를 만들어, 모든 파일을 해당 폴더에 넣고 압축  
(ex) Lab1\_2018147000\_김영희.zip
- 보고서 파일 이름은 "Lab1\_학번\_이름.pdf"으로 작성하며, 반드시 pdf 파일로 작성  
(ex) Lab1\_2018147000\_홍길동.pdf
- module AddSub(A, B, op, carry, S, overflow) 은 하나의 모듈로 구현하거나 여러 개의 submodule로 나누어 구현하여도 무방함
- verilog 소스 파일 이름은 "모듈명.v" 으로 작성하며, 하나의 verilog 파일에 하나의 모듈만 구현  
(ex) AddSub.v
- 한글로 주석 작성 시 ModelSim 에서 문자열 인코딩 에러가 발생하므로, 소스 코드 설명은 보고서에 작성
- 보고서는 1) 모듈 구현에 대한 전체적 설명 (**Figure 1** 참조), 2) sample input 에 대해 시뮬레이션을 수행한 후의 waveform 결과 화면 등을 반드시 포함하며, 3) 보고서 내용은 자유 형식으로 기술하되 4) 파일 형식은 pdf 파일로 제출

## 채점 기준

---

- **module interface :**
  - ✓ 정해진 input 을 입력하면 몇 개의 올바른 output 이 나오는가?  
(주어진 sample input 은 모두 넣어볼 예정)
  - ✓ 모듈의 input/output port 이름 및 순서가 지켜졌는가?  
(지켜지지 않으면 testbench 가 제대로 동작 안 함)
- **보고서 내용 :**
  - ✓ 모듈 동작에 대한 전체적 설명, 모듈 구현에 대한 개략적 설명, sample input/output 에 대한 waveform 등
- **과제 제출 형식 :**
  - ✓ 명시된 과제 제출 형식을 잘 지켰는가?
  - ✓ 경로, 파일 형식 등  
(현재 과제는 HW2 이지만, 경로 및 보고서 이름은 "Lab1\_학번\_이름" 및 "Lab1\_학번\_이름.pdf" 으로 작성 부탁드립니다)

## Reference

---

- Navabi, Zainalabedin. Verilog digital system design. McGraw-Hill, 1999.
- 이문기, "HDL을 이용한 디지털 논리 실험". 홍릉과학출판사, 2004  
<http://library.yonsei.ac.kr/search/detail/CAT000000096890>  
Chapter 07. 4-비트 Adder/Subtractor