

Adder Subtractor

Data input: 2 N-bit vectors x & y

Data output: 1 N-bit vector $sumDiff$

Control: 1-bit $func$

Status: 2-bits $sovf$ $uovf$

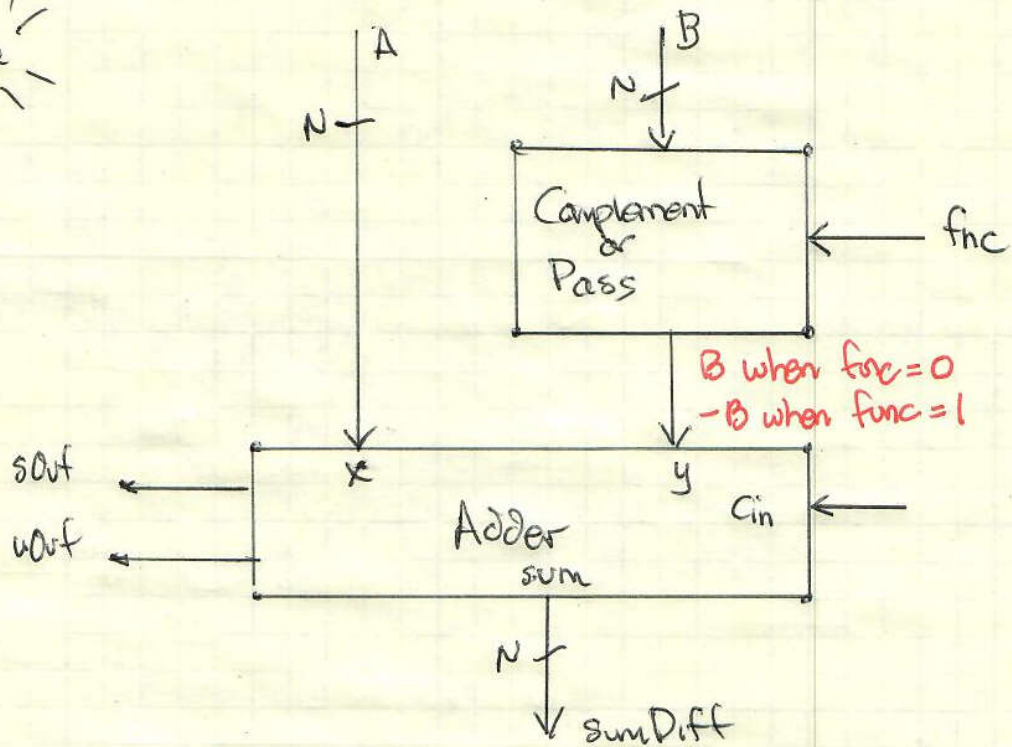
Behavior:

$$sumDiff = \begin{cases} x+y & \text{when } func=0 \\ x-y & \text{when } func=1 \end{cases}$$

$sovf = 1$ when signed overflow

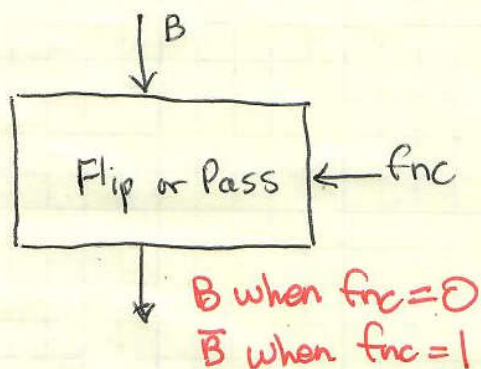
$uovf = 1$ when unsigned overflow

Idea

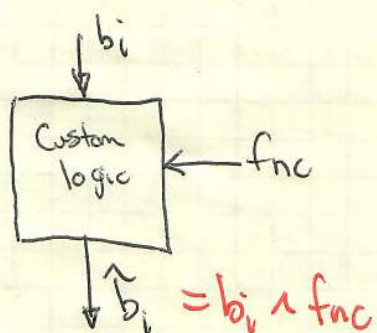


Idea

Split complement into 2 operations
 Flip bits Add 1
 Custom Logic Use Cin of adder



Bit Slice



b_i	fnc	\hat{b}_i
0	0	0
0	1	1
1	0	0
1	1	1

Comparators :

Data in : 2 N-bit vectors X, Y interpreted as binary numbers
 Data out : None
 Control : None
 Status : 3-bits GT, LT, EQ

Behavior :
 $GT = \begin{cases} 1 & \text{if } X > Y \\ 0 & \text{otherwise} \end{cases}$
 $LT = \begin{cases} 1 & \text{if } X < Y \\ 0 & \text{otherwise} \end{cases}$
 $EQ = \begin{cases} 1 & \text{if } X == Y \\ 0 & \text{otherwise} \end{cases}$

Internal organization of 4-bit comparator

