```
MODULE      addsub
TITLE      'Adder/Subtracter with Flags'


" addsub        DEVICE  'MACH4032'


" Description:  This is the program for Homework #4. It implements a
"               4-bit Adder/Subtracter which includes flags summarizing the
"               result (zero, carry/borrow, overflow, and sign). It also has
"               outputs that output the comparison result (assuming a
"               subtraction is being done).


" Revision History:
" 04/24/07   Glen George  Initial Revision
" 04/25/07   Glen George  Fixed logic so add and subtract both work and fit
" 04/25/07   Glen George  Fixed reversal of SignedGT and SignedLT outputs
"                         (label on PCB is wrong)
" 10/24/07   Glen George  Changed pinout and Subtract polarity to match new
"                         PCB
" 10/24/07   Glen George  Changed file to match the new assignment
" 10/24/07   Glen George  Updated comments
" 11/02/08   Glen George  Updated comments
" 01/12/18   Glen George  Changed pinout to match new PCB (v3.0)
" 01/04/21   Glen George  Updated comments
" 02/03/23   Steven Lei   Updated description
" 02/03/23   Steven Lei   Added equations for adder/subtracter and flags
" 02/03/23   Steven Lei   Added comments explaining code

" Pins

" Inputs

Subtract        pin 37;                 "input   Add/Subtract input (low for add)


" Operand A input
A3              pin 31;                 "input   operand A, bit 3
A2              pin 30;                 "input   operand A, bit 2
A1              pin 29;                 "input   operand A, bit 1
A0              pin 26;                 "input   operand A, bit 0

" Operand B input
B3              pin 22;                 "input   operand B, bit 3
B2              pin 21;                 "input   operand B, bit 2
B1              pin 20;                 "input   operand B, bit 1
B0              pin 19;                 "input   operand B, bit 0
```

```
" Sum/Difference output

Sum4            pin 13   ISTYPE 'COM';  "output  Sum/Difference, bit 4
Sum3            pin  3   ISTYPE 'COM';  "output  Sum/Difference, bit 3
Sum2            pin  2   ISTYPE 'COM';  "output  Sum/Difference, bit 2
Sum1            pin 44   ISTYPE 'COM';  "output  Sum/Difference, bit 1
Sum0            pin 43   ISTYPE 'COM';  "output  Sum/Difference, bit 0


" Flag outputs

Carry           pin  9   ISTYPE 'COM';  "output  Carry flag
Overflow        pin  8   ISTYPE 'COM';  "output  Overflow flag
Sign            pin  7   ISTYPE 'COM';  "output  Sign flag
Zero            pin  4   ISTYPE 'COM';  "output  Zero flag


" Comparison outputs

UnsignedGT      pin 15   ISTYPE 'COM';  "output  unsigned greater than (>)
UnsignedEQ      pin 41   ISTYPE 'COM';  "output  unsigned equal to     (=)
UnsignedLT      pin 24   ISTYPE 'COM';  "output  unsigned less than    (<)
SignedGT        pin 16   ISTYPE 'COM';  "output  signed greater than (>)
SignedEQ        pin 42   ISTYPE 'COM';  "output  signed equal to     (=)
SignedLT        pin 25   ISTYPE 'COM';  "output  signed less than    (<)


" Unconnected Pins

"IOA13          pin 14                  input/output
"IOB0           pin 18                  input/output
"IOB12          pin 35                  input/output
"IOB13          pin 36                  input/output

"IOGOE0         pin 40                  input/output/output enable
"IOGOE1         pin 38                  input/output/output enable
"ICLK0          pin 39                  input/clock
"ICLK2          pin 17                  input/clock


" Programming Pins (not available for use in the design)

"TCK            pin 10                  pgm     programming interface TCK
"TDI            pin  1                  pgm     programming interface TDI
"TDO            pin 32                  pgm     programming interface TDO
"TMS            pin 23                  pgm     programming interface TMS


" Power Pins
```

```
"GND            pin  5                  supply  power ground
"GND            pin 12                  supply  power ground
"GND            pin 27                  supply  power ground
"GND            pin 34                  supply  power ground
"VCC            pin 11                  supply  power Vcc
"VCC            pin 33                  supply  power Vcc
"VCCIO          pin 28                  supply  power Vcc I/O
"VCCIO          pin  6                  supply  power Vcc I/O


" Intermediate Terms

CarryOut0       node     ISTYPE 'COM';  "node    carry out of bit 0
CarryOut1       node     ISTYPE 'COM';  "node    carry out of bit 1
CarryOut2       node     ISTYPE 'COM';  "node    carry out of bit 2


EQUATIONS

" The equations for each sum bit and carry bit are generated by implementing
" the add/ subtracter design in HW3. The B inputs must be XOR'd with subtract
" to account for subtraction. The carry in for the 0 bit is just subtract.

Sum0 = (A0 $ (B0 $ Subtract)) $ Subtract;
Sum1 = (A1 $ (B1 $ Subtract)) $ CarryOut0;
Sum2 = (A2 $ (B2 $ Subtract)) $ CarryOut1;
Sum3 = (A3 $ (B3 $ Subtract)) $ CarryOut2;

CarryOut0 = (A0 & (B0 $ Subtract)) # ((A0 $ (B0 $ Subtract)) & Subtract);
CarryOut1 = (A1 & (B1 $ Subtract)) # ((A1 $ (B1 $ Subtract)) & CarryOut0);
CarryOut2 = (A2 & (B2 $ Subtract)) # ((A2 $ (B2 $ Subtract)) & CarryOut1);

" Note that the last sum bit is the carryout of the previous sum bit.

Sum4 = (A3 & (B3 $ Subtract)) # ((A3 $ (B3 $ Subtract)) & CarryOut2);

" FLAGS

" The flags describe the output of the adder/subtracter.
" Zero = if the sum/difference is 0
" Carry = carryout of the last bit (sum4)
" Sign = sign bit of a signed number
" Overflow = if the sum/difference is beyond the bounds (0,15) or (-8,7)

Zero = (!Sum0 & !Sum1 & !Sum2 & !Sum3);

" The carry is the last carryout bit (Sum4). But since the signal
" needs to be active high carry/borrow it needs to be XOR'd with
" subtract for correction.
```

```
Carry = Sum4 $ Subtract;

Sign = Sum3;

" The overflow equation can be solved by treating all
" cases as addition (since substraction is the same as
" adding the negative). Overflow occurs in addition
" either when pos + pos = neg or neg + neg = pos.

Overflow = (Sum4 & !CarryOut2) # (!Sum4 & CarryOut2);


" Unsigned comparisons using flags only (assuming subtraction)

UnsignedEQ = Zero;

" For UnsignedGT, there should be no carry (overflow applies only for signed)
" if A > B (all solutions are between A-B=0 and A-B = A)

UnsignedGT = (!Carry & !Zero);

" less than is just not greater than or equal
UnsignedLT  =  !(UnsignedGT # UnsignedEQ);


" Signed comparisons using flags only (assuming subtraction)

SignedEQ = Zero;

" For SignedGT, there are three comparison cases to consider:
" for positive A, positive B -> no overflow, no carry, no sign
" for positive A, negative B, -> could be overflow, always carry, sign depends on
overflow
" for negative A, negative B, -> no overflow, no carry, no sign

SignedGT = !Zero & ((!Carry & !Sign & !Overflow) # (Carry & !(Sign $ Overflow)));

" less than is just not greater than or equal
SignedLT  =  !(SignedGT # SignedEQ);


END  addsub
```