COMP2611: Computer Organization

Arithmetic Logic Unit

Arithmetic Logic Unit

Review of the 1-bit ALU

- 1-bit ALU
- 1-bit ALU for the MSB
- Overflow detection

An extended 32-bit ALU

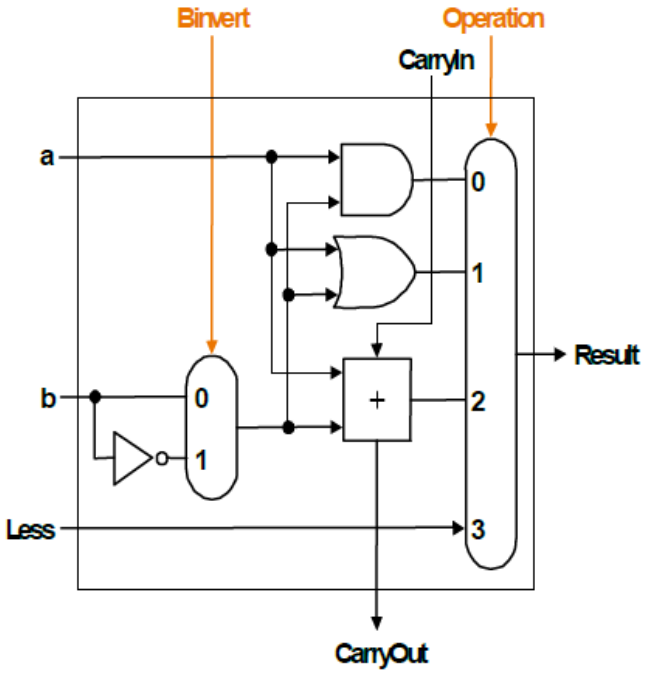
- Ripple carry Add/Sub
- SLT implementation

Exercises

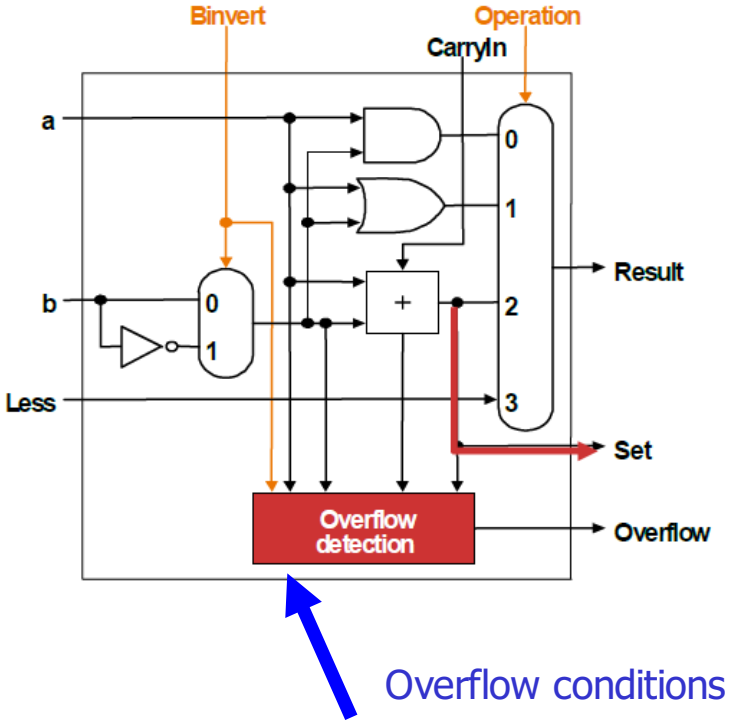
1-bit ALUs

A 32-bit ALU can be constructed using the following 1-bit ALUs

- 1-bit ALU for bits 0 to 30



- 1-bit ALU for the Most Significant Bit (MSB):



Overflow conditions

Operation	Sign Bit of X	Sign Bit of Y	Sign Bit of Result
X + Y	0	0	1
X + Y	1	1	0
X – Y	0	1	1
X – Y	1	0	0

Arithmetic Logic Unit

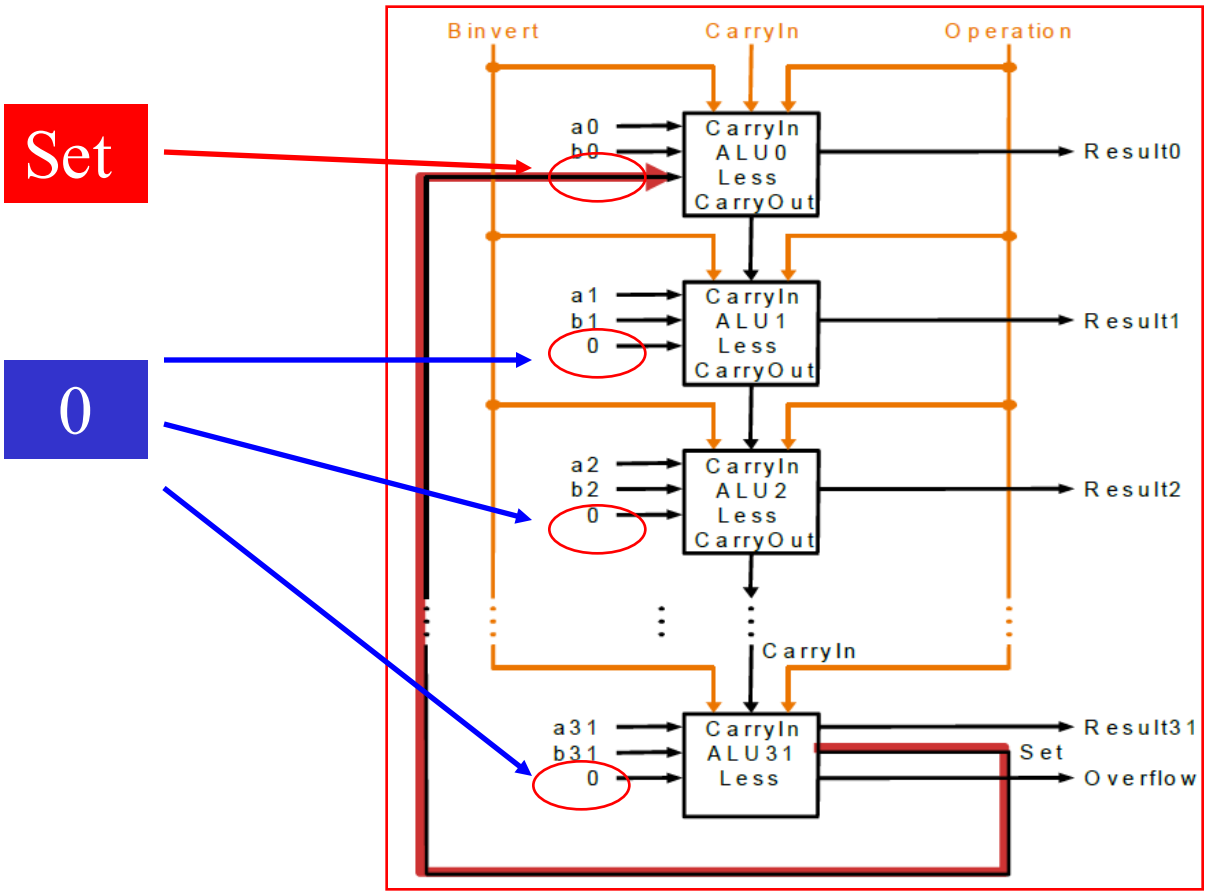
Review of the 1-bit ALU
- 1-bit ALU
- ALU for the MSB
- Overflow detection

An extended 32-bit ALU
- Ripple carry Add/Sub
- SLT implementation

Exercises

A 32-bit ALU that supports ADD, SUB, AND, OR, SLT

- An extended 32-bit ALU (supports SLT) can be formed by connecting 32 1-bit ALUs as follows. Note the 0's at the "Less" input for ALU1-ALU31, note also the set signal from ALU31 to ALU0.



Arithmetic Logic Unit

Review of the 1-bit ALU
- 1-bit ALU
- ALU for the MSB
- Overflow detection

An extended 32-bit ALU
- Ripple carry Add/Sub
- SLT implementation

Exercises

Exercises

Question 1: Some argue that the control signals Binvert and CarryIn of the bit-0 ALU can be combined into one control signal. Justify this claim (refer to the ALU diagrams on slides 3 and 4 whenever necessary).

Solution:

Addition operation : both Binvert and CarryIn of the bit-0 ALU should be 0.

Subtraction operation: both Binvert and CarryIn of the bit-0 ALU should be 1.

OR/AND operations : Binvert should be 0, CarryIn should be "don't care".

Combining the two signals does not impair their functions. Therefore it is okay to combine both signals into one. In fact they are combined to form the Bnegate signal.

Question 2: By referring to slides 3 and 4, explain how SLT operation can be perfromed. State the values for the control signals Binvert, CarryIn and Operation.

Solution:

SLT outputs an "1" when the upper operand A is less than the lower operand B. The subtraction operation A-B will be performed.

When A-B<0, the sign bit (result of the MSB) will be 1 and will be forwarded to ALU0 (so "Less" becomes 00…01) .

When A-B>=0, "Less" will be 00…00.

The signals Binvert and CarryIn of ALU0 should be set to "1" to enable the subtraction, the signal Operation should be set to 3 ($11_{(2)}$) to enable the resulting "set" to be forwarded to the output.

Question 3: By referring to slides 3 and 4,derive the logic expression in the Sum of Product form (SoP) for overflow conditions.

Solution: Two types of overflows according to the table below,

1) addition overflow Binvert=0, a3=b3=0, set=1 or
Binvert=0, a3=b3=1, set=0

2) subtraction overflow Binvert=1, a3=0,b3=1, set=1 or
Binvert=1, a3=1,b3=0, set=0

Operation	Sign Bit of X	Sign Bit of Y	Sign Bit of Result
X + Y	0	0	1
X + Y	1	1	0
X – Y	0	1	1
X – Y	1	0	0

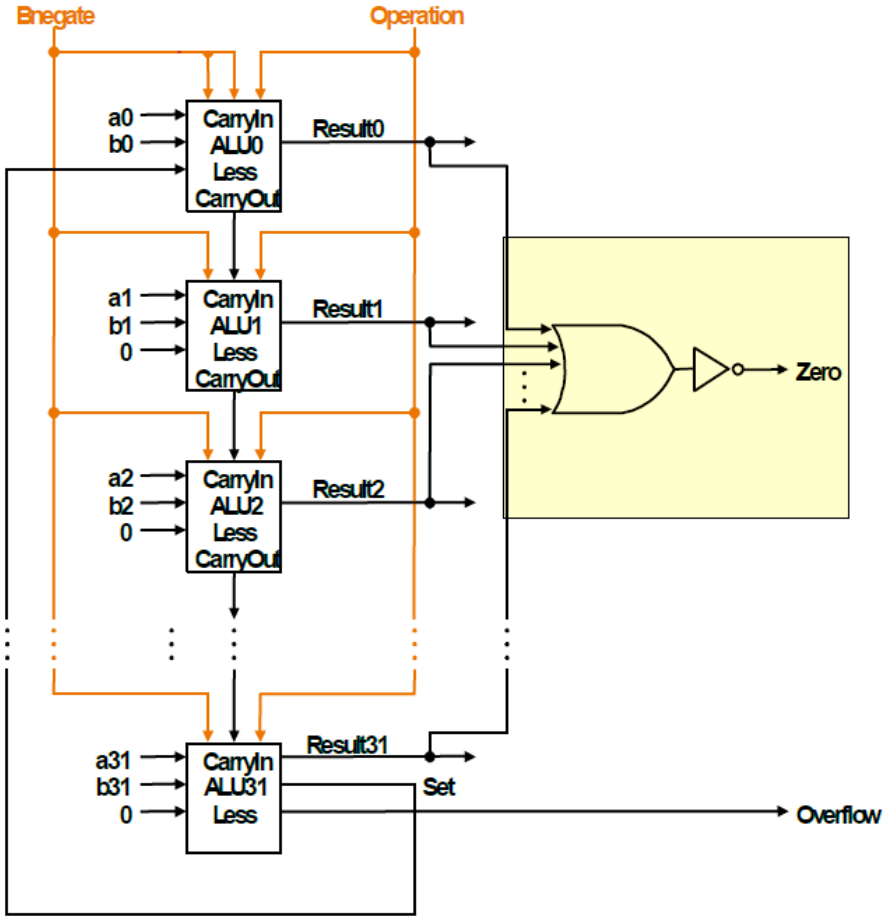
The corresponding SoP is:

$$\overline{\text{Binvert}}\cdot\overline{\text{a3}}\cdot\overline{\text{b3}}\cdot\text{set}+\overline{\text{Binvert}}\cdot\text{a3}\cdot\text{b3}\cdot\overline{\text{set}}+\text{Binvert}\cdot\overline{\text{a3}}\cdot\text{b3}\cdot\text{set}+\text{Binvert}\cdot\text{a3}\cdot\overline{\text{b3}}\cdot\overline{\text{set}}$$

Question 4: The SLT operation depends on the result of A-B, and set whenever the sign bit of the operation is asserted. Describe a scenario such that this approach does not work correctly.

Solution: When the subtraction overflows, this mechanism does not work correctly. To see this, assume A>0, B<0, when A-B overflows, (result3 equals to 1 then) the mechanism will consider A less than B.

Question 5: By referring to the modified 32-bit ALU below, explain how the condition A==B is detected. State the values for the control signals Bnegate and Operation.



Exercises

Solution: To check A==B, we perform the subtraction A-B, if the result is 0 (i.e. result0=...=result31=0) then A==B. The NOR gate in the figure will output 1 iff all the result bits are 0. Thus if the NOR gate outputs 1, then A==B. To perform the subtraction, Bnegate is set to 1 and Operation is set to 10.