COMP2611: Computer Organization

Arithmetic Logic Unit

Overview

- You will learn the following in this tutorial:
  - the structure of 1-bit MIPS ALUs,
  - overflow detecting mechanism in the MSB 1-bit ALU,
  - building a simple 32-bit ALU for the MIPS arithmetic,
  - implementing the SLT instruction in a 32-bit MIPS ALU.

Arithmetic Logic Unit

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

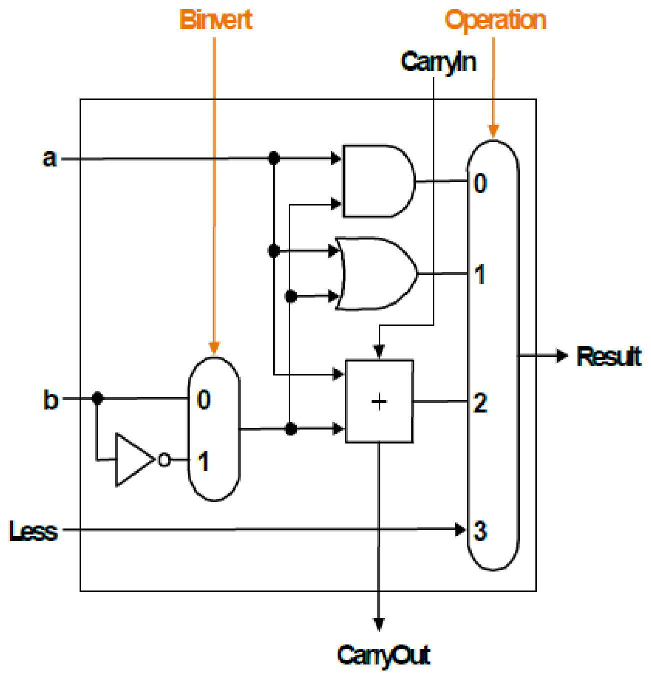
An extended 32-bit ALU
- Ripple carry Add/Sub
- SLT implementation

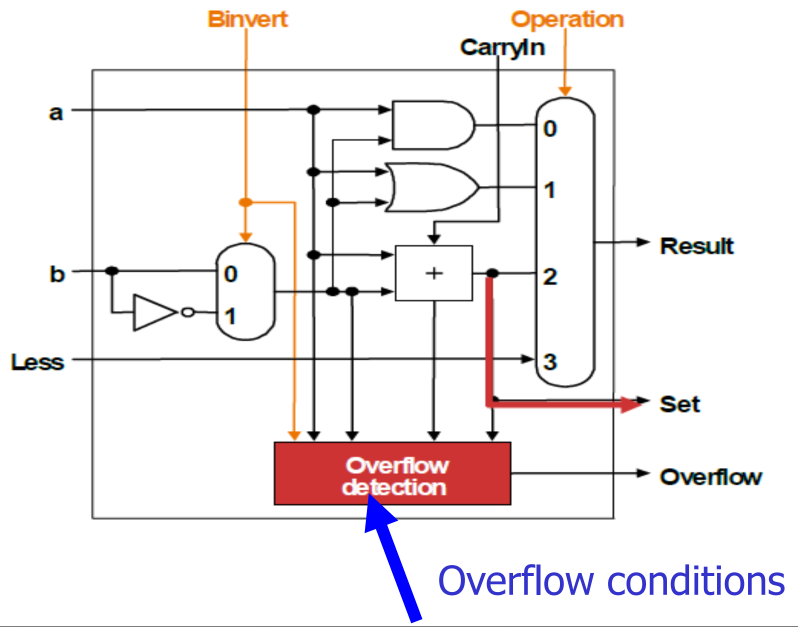
Exercises

1-bit ALU0-ALU30 V.S. 1-bit ALU31

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): ALU31





Operation	Sign Bit of A	Sign bit of B	Sign bit of Result
A + B	0	0	1
A + B	1	1	0
A - B	0	1	1
A - B	1	0	0

Arithmetic Logic Unit

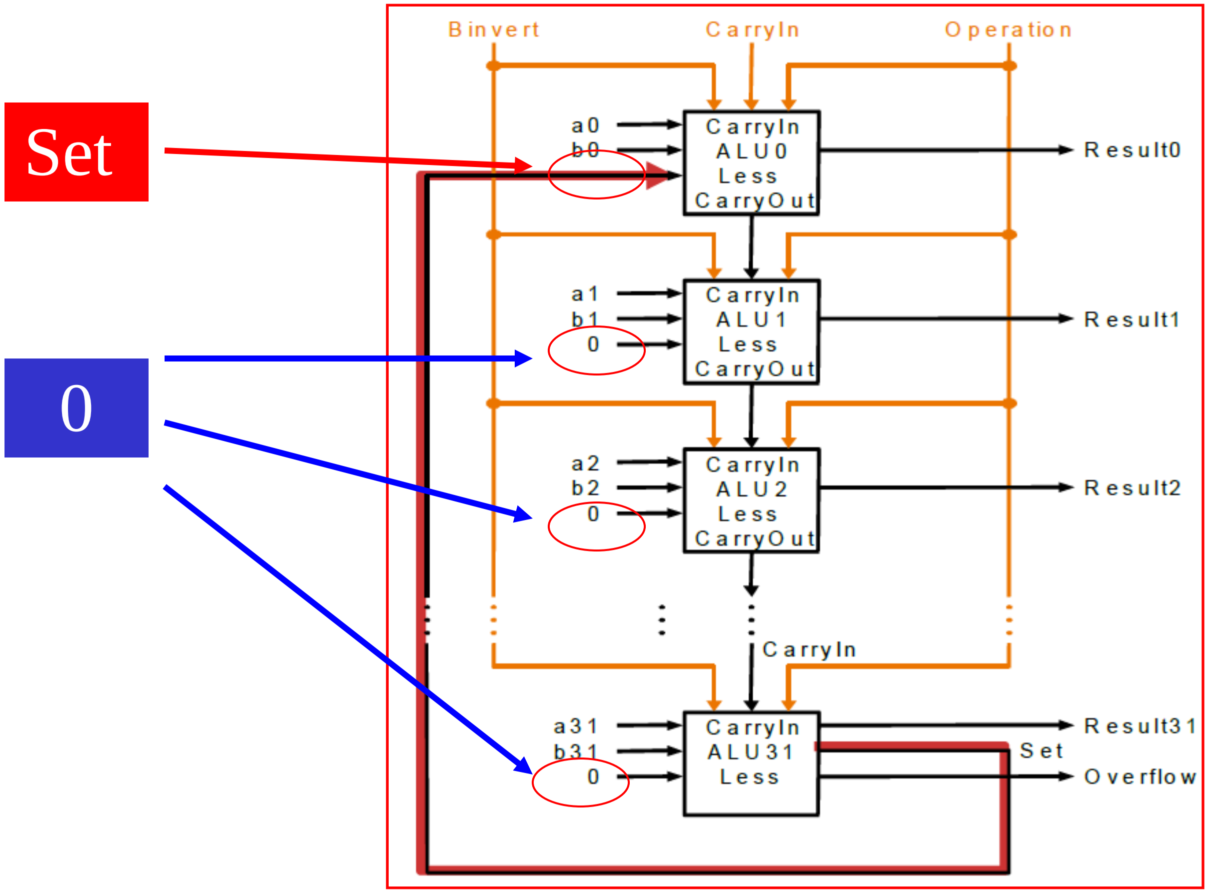
Review of the 1-bit ALU
- 1-bit ALU
- ALU for the MSB
- Overflow detection

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

Exercises

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

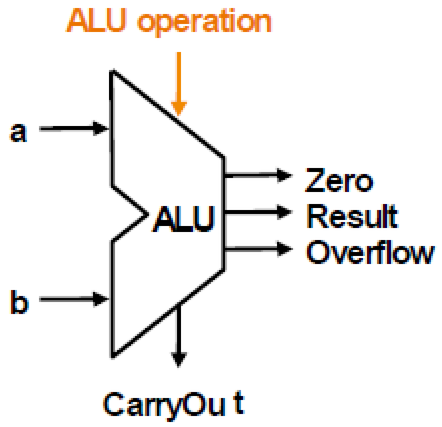
- An extended 32-bit ALU (supports SLT) can be formed by connecting 32 1-bit ALUs as follows. Note the 0's at the "Less" input for ALU1-ALU31, note also the set signal from ALU31 to ALU0.



ALU Control Signals /Control Lines

	Binvert	CarryIn	Operation
AND	0	X	00
OR	0	X	01
ADD	0	0	10
SUB	1	1	10
SLT	1	1	11

Control signals to ALU0



Bnegate (combine Binvert and CarryIn)

Operation	ALU Control Lines
AND	000
OR	001
ADD	010
SUB	110
SLT	111

Final ALU control lines

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 4 and 6 whenever necessary).

Exercises

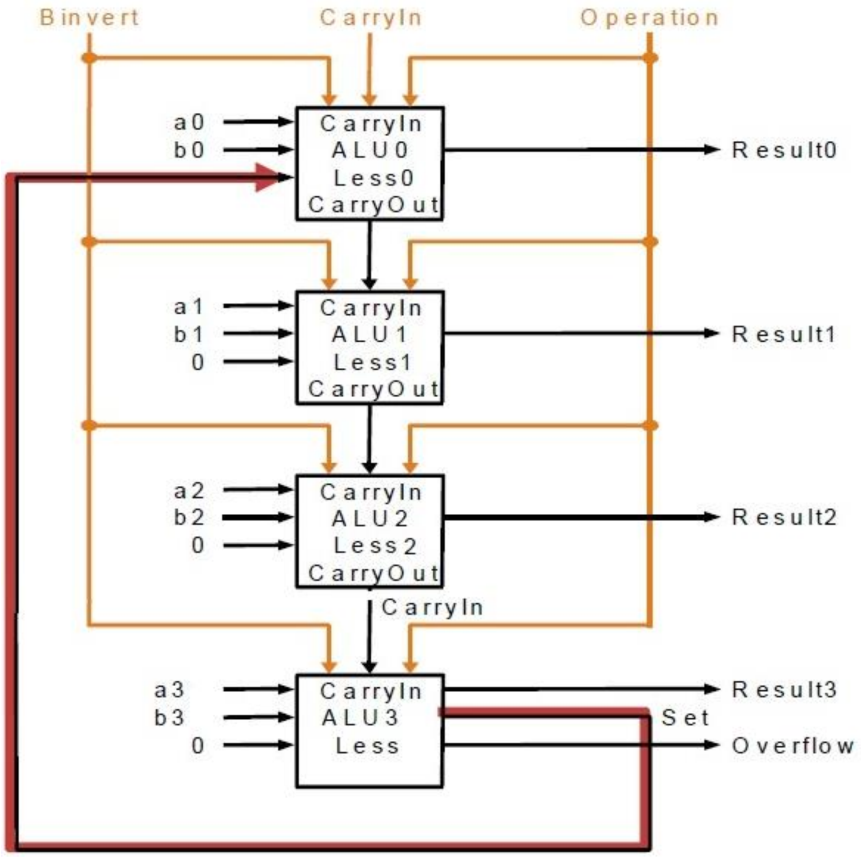
Question 2: By referring to slides 4 and 6, explain how SLT operation can be performed. State the values for the control signals Binvert, CarryIn and Operation.

Exercises

Question 3: By referring to slides 4 and 6, derive the logic expression in the Sum of Product form (SoP) for overflow conditions based on a 4-bit ALU.

Exercises

- Question 4: By referring to the 4-bit ALU below, and by filling into the blanks in the table on the next slide, show how the SLT operation is performed. Assume A is 4-bit and is equal to -3, B is also 4-bit and equals to 2.



Exercises

Solution:

ALU0					Result of Operation			
a0	b0	CarryIn	Binvert	CarryOut	0	1	2	3

ALU1					Result of Operation			
a1	b1	CarryIn	Binvert	CarryOut	0	1	2	3

ALU2					Result of Operation			
a2	b2	CarryIn	Binvert	CarryOut	0	1	2	3

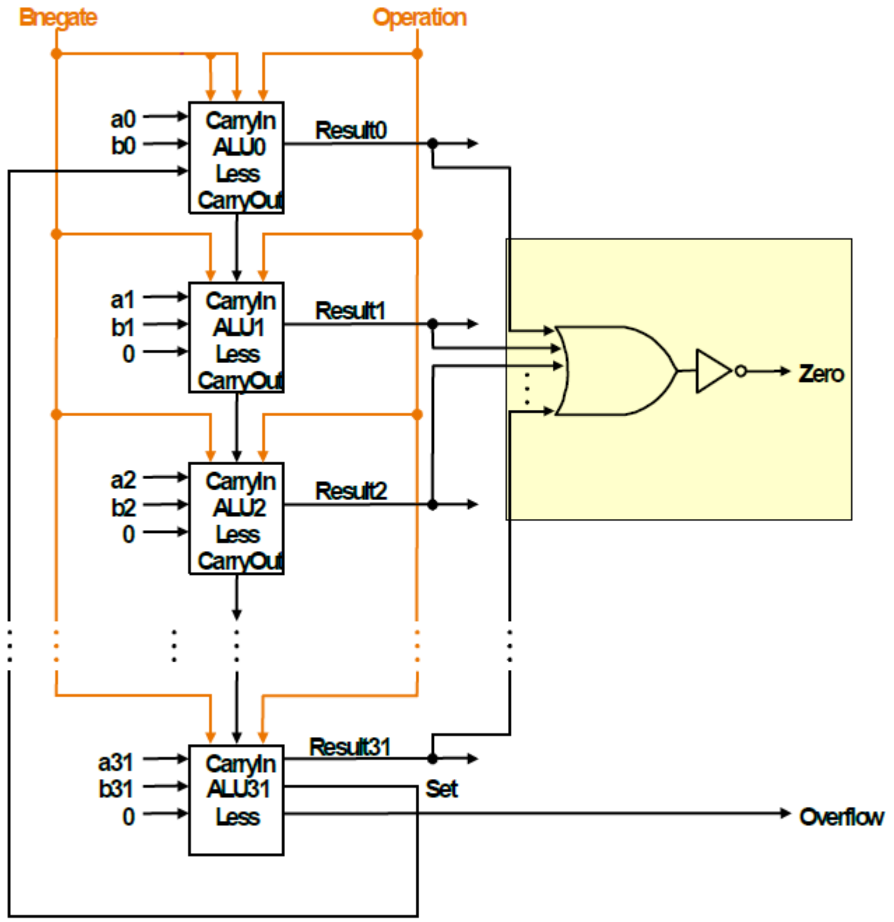
ALU3					Result of Operation			
a3	b3	CarryIn	Binvert	CarryOut	0	1	2	3

Exercises

Question 5: 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.

Exercises

Question 6: 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.



Conclusion

- Today we have reviewed:
  - the structure of 1-bit MIPS ALUs,
  - overflow detecting mechanism in the MSB 1-bit ALU,
  - a simple 32-bit ALU for the MIPS arithmetic,
  - implementing the SLT instruction in a 32-bit MIPS ALU,
  - implementing the "zero" testing in a 32-bit ALU.