Processor: Data Path Components

Abstraction!

Arithmetic Logic Unit (ALU) ①

Hardware unit for arithmetic and bitwise operations.

1-bit ALU for bitwise operations
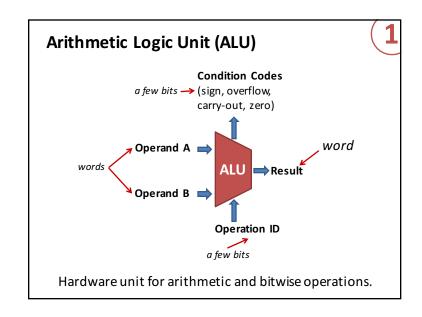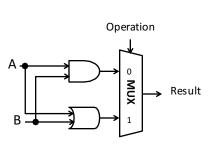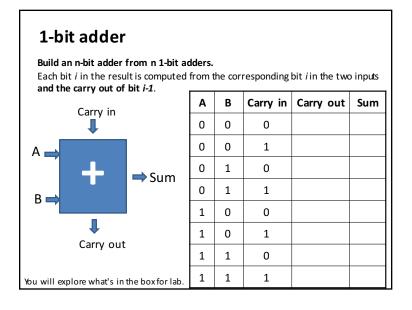
Build an n-bit ALU from n 1-bit ALUs.
Each bit i in the result is computed from the corresponding bit i in the two inputs.

Op	A	B	Result
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

1-bit adder

Build an n-bit adder from n 1-bit adders.
Each bit i in the result is computed from the corresponding bit i in the two inputs **and the carry out of bit $i-1$**.

A	B	Carry in	Carry out	Sum
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

You will explore what's in the box for lab.

1-bit ALU

Operation

Carry in

a

b

MUX → Result

Sum

Carry out

32-bit ALU with ripple carry

Operation

CarryIn

a0, b0 → CarryIn ALU0 CarryOut → Result0

a1, b1 → CarryIn ALU1 CarryOut → Result1

a2, b2 → CarryIn ALU2 CarryOut → Result2

a31, b31 → CarryIn ALU31 → Result31

Adding subtraction

invert B

Carry in

Operation

A

B

MUX

0
MUX
1

Sum

+

MUX → Result

Carry out

ALU Condition Codes

Zero Flag: 1 if result is 00...0 else 0

Sign Flag: sign bit of result

Carry Flag: 1 if unsigned overflow else 0
carry-out bit of result

Overflow Flag: 1 if signed overflow else 0

Compute NOT, NAND, NOR, XOR, <, ==, …?
Detect overflow?

invert A
invert B
Carry in
Operation
2

A
0
MUX
1

MUX
Result

B
0
MUX
1

+ Sum

0
1
2

Carry out

Controlling the ALU

Abstraction!

ALU control lines	Function
0000	AND
0001	OR
0010	add
0110	subtract
1100	NOR

Invert A Invert B Operation ID

Condition Codes

Operand A

ALU

Operand B

Result

Control Lines