Digital Design and Computer Organization

Adder, Subtractor, Overflow - 3

Reetinder Sidhu

Department of Computer Science and Engineering

Digital Design and Computer Organization

Adder, Subtractor, Overflow - 3

Reetinder Sidhu

Department of Computer Science and Engineering

Adder, Subtractor, Overflow - 3

Course Outline

- Digital Design
  - Combinational logic design
    - **Adder, Subtractor, Overflow - 3**
  - Sequential logic design
- Computer Organization
  - Architecture (microprocessor instruction set)
  - Microarchitecure (microprocessor operation)

Concepts covered

- Logic Circuits for:
  - Increment
  - Two's Complement
  - Subtractor
  - Adder / Subtractor

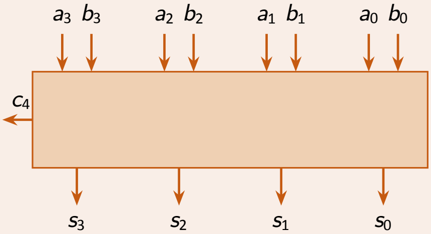
Adder, Subtractor, Overflow - 3

Increment Logic Circuit

Adder, Subtractor, Overflow - 3

Increment Logic Circuit

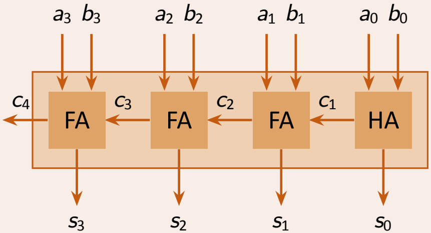
4-bit Ripple Carry Adder



Adder, Subtractor, Overflow - 3

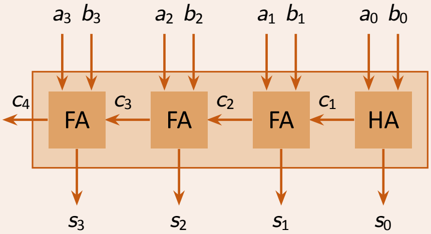
Increment Logic Circuit

4-bit Ripple Carry Adder



Increment Logic Circuit

4-bit Ripple Carry Adder

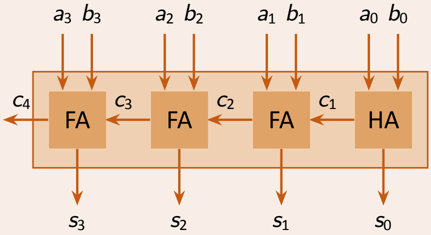


- Carry “ripples” through the cary chain

Adder, Subtractor, Overflow - 3

Increment Logic Circuit

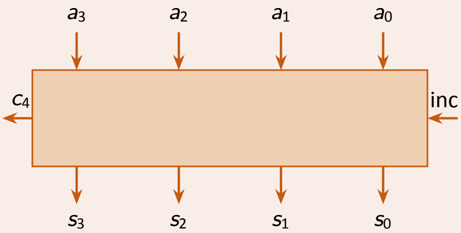
4-bit Ripple Carry Adder



- Carry “ripples” through the cary chain

Incrementer Logic Circuit

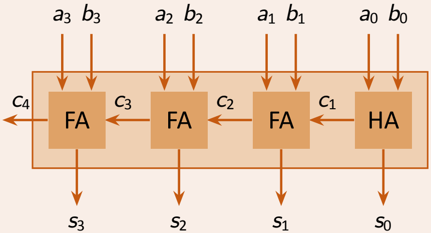
- Add 1 to input number



Adder, Subtractor, Overflow - 3

Increment Logic Circuit

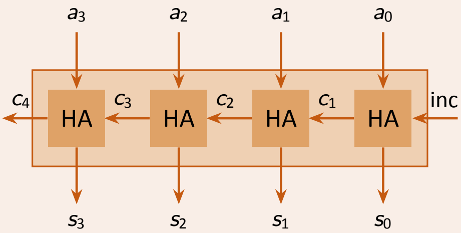
4-bit Ripple Carry Adder



- Carry “ripples” through the cary chain

Incrementer Logic Circuit

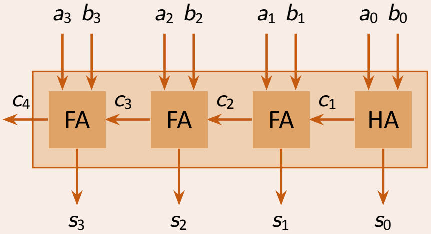
- Add 1 to input number



Adder, Subtractor, Overflow - 3

Increment Logic Circuit

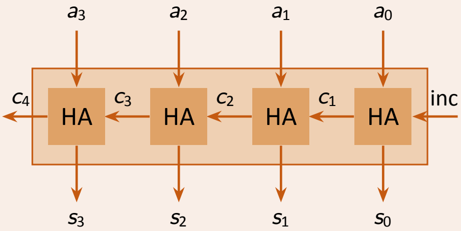
4-bit Ripple Carry Adder



- Carry “ripples” through the cary chain

Incrementer Logic Circuit

- Add 1 to input number



- Less logic resources than ripple carry adder

Two’s Complement Subtractor

Two’s Complement Subtractor

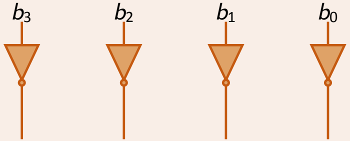
- Subtract one four bit number ($b_3\,b_2\,b_1\,b_0$) from another ($a_3\,a_2\,a_1\,a_0$)
- Take two’s complement of $b_3\,b_2\,b_1\,b_0$:
  - Invert $b_3\,b_2\,b_1\,b_0$
  - Add one to it
- Add above number to $a_3\,a_2\,a_1\,a_0$ yielding result

Adder, Subtractor, Overflow - 3

Two's Complement Subtractor

Two's Complement Subtractor

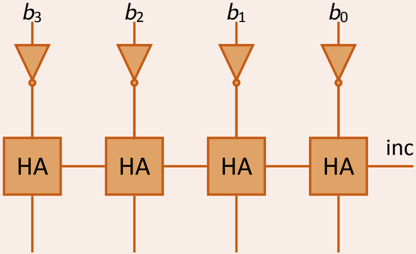
- Subtract one four bit number ($b_3\,b_2\,b_1\,b_0$) from another ($a_3\,a_2\,a_1\,a_0$)
- Take two's complement of $b_3\,b_2\,b_1\,b_0$:
  - Invert $b_3\,b_2\,b_1\,b_0$
  - Add one to it
- Add above number to $a_3\,a_2\,a_1\,a_0$ yielding result



Adder, Subtractor, Overflow - 3
Two's Complement Subtractor

Two's Complement Subtractor

- Subtract one four bit number ($b_3\,b_2\,b_1\,b_0$) from another ($a_3\,a_2\,a_1\,a_0$)
- Take two's complement of $b_3\,b_2\,b_1\,b_0$:
  - Invert $b_3\,b_2\,b_1\,b_0$
  - Add one to it
- Add above number to $a_3\,a_2\,a_1\,a_0$ yielding result

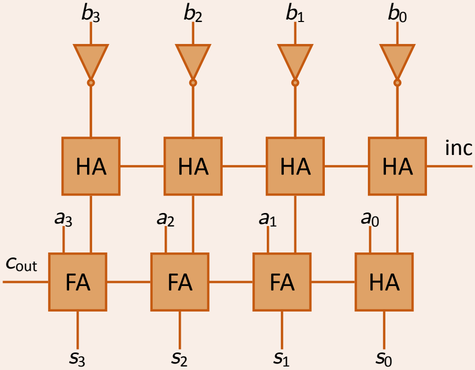


ADDER, SUBTRACTOR, OVERFLOW - 3

Two's Complement Subtractor

Two's Complement Subtractor

- Subtract one four bit number ($b_3 b_2 b_1 b_0$) from another ($a_3 a_2 a_1 a_0$)
- Take two's complement of $b_3 b_2 b_1 b_0$:
  - Invert $b_3 b_2 b_1 b_0$
  - Add one to it
- Add above number to $a_3 a_2 a_1 a_0$ yielding result



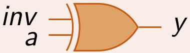
Two's Complement Adder / Subtractor

XOR as Controlled Inverter

- Truth table:

inv	*a*	*y*
0	0	0
0	1	1
1	0	1
1	1	0

- Symbol:



- When $inv = 0, y = a$
- When $inv = 1, y = \overline{a}$

Adder, Subtractor, Overflow - 3

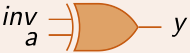
Two's Complement Adder / Subtractor

XOR as Controlled Inverter

- Truth table:

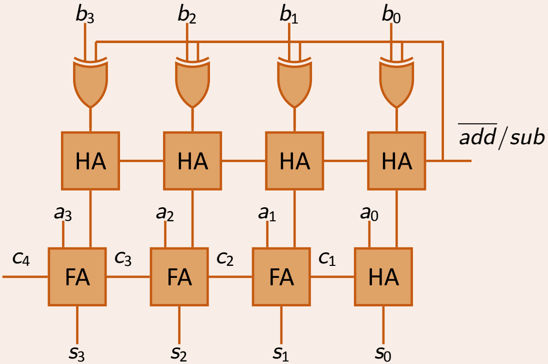
inv	*a*	*y*
0	0	0
0	1	1
1	0	1
1	1	0

- Symbol:



- When $inv = 0$, $y = a$
- When $inv = 1$, $y = \overline{a}$

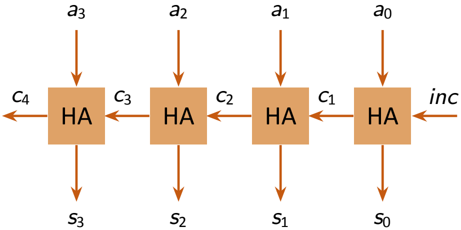
Two's Complement Adder / Subtractor



- $\overline{add}/sub$ denotes:
  - Addition when $\overline{add}/sub = 0$
  - Subtraction when $\overline{add}/sub = 1$

ADDER, SUBTRACTOR, OVERFLOW - 3

Think About It



- Above logic circuit increments only when $inc = 1$
- Construct a logic circit which always increments (so no *inc* input)
- Construct a decrementer logic circuit