CSE211

Computer Organization and Design

Lecture : 3 Tutorial: 1 Practical: 0 Credit: 4

MICROOPERATIONS

Computer system microoperations are of four types:

- **Arithmetic microoperations**
- **Register transfer microoperations**
- **Logic microoperations**
- **Shift microoperations**

Arithmetic MICROOPERATIONS

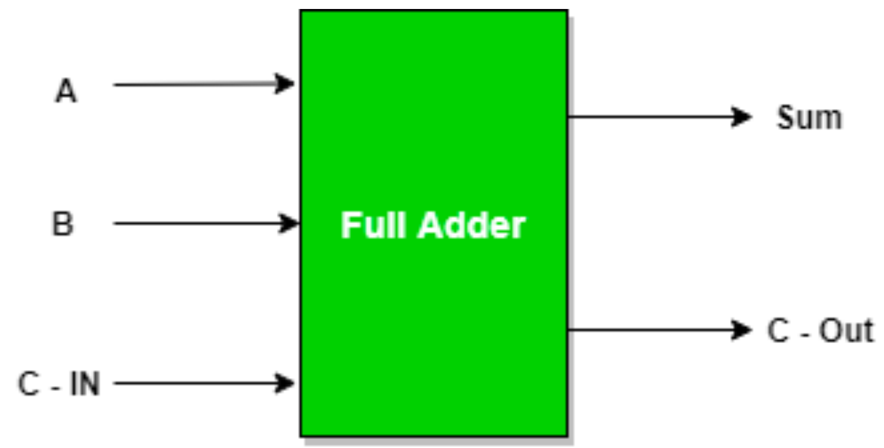
- **The basic arithmetic microoperations are**
  - Addition
  - Subtraction
  - Increment
  - Decrement

- **The additional arithmetic microoperations are**
  - Add with carry
  - Subtract with borrow
  - Transfer/Load
  - etc. …

Summary of Typical Arithmetic Micro-Operations

R3 ← R1 + R2	Contents of R1 plus R2 transferred to R3
R3 ← R1 - R2	Contents of R1 minus R2 transferred to R3
R2 ← R2’	Complement the contents of R2
R2 ← R2’+ 1	2's complement the contents of R2 (negate)
R3 ← R1 + R2’+ 1	subtraction
R1 ← R1 + 1	Increment
R1 ← R1 - 1	Decrement

Full Adder is the adder which adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM.



Inputs			Outputs	
A	B	C – IN	Sum	C - Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Binary Adder

4-bit Binary Adder : *Fig. 4-6*

- Full adder = 2-bits sum + previous carry
- Binary adder = the arithmetic sum of two binary numbers of any length
- c_0(input carry), c_4(output carry)

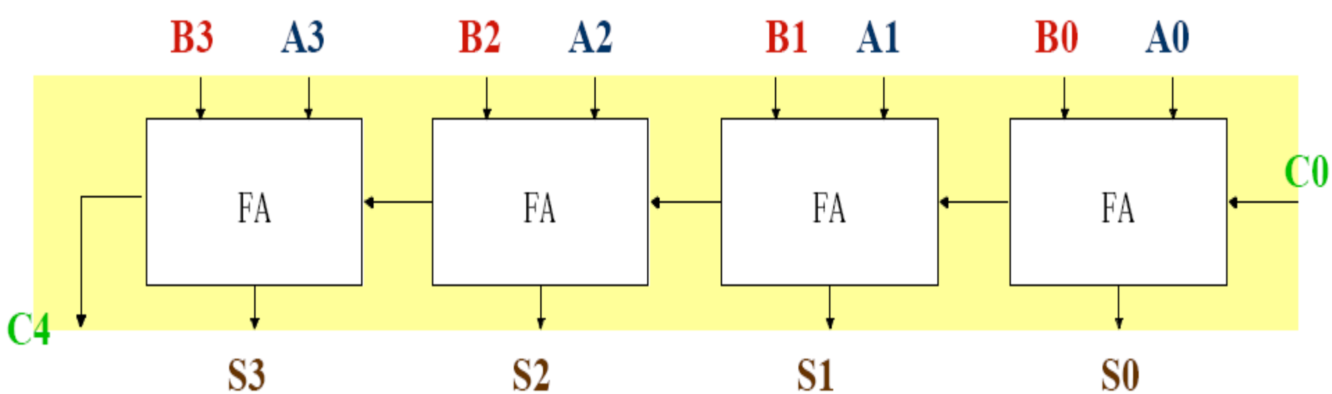
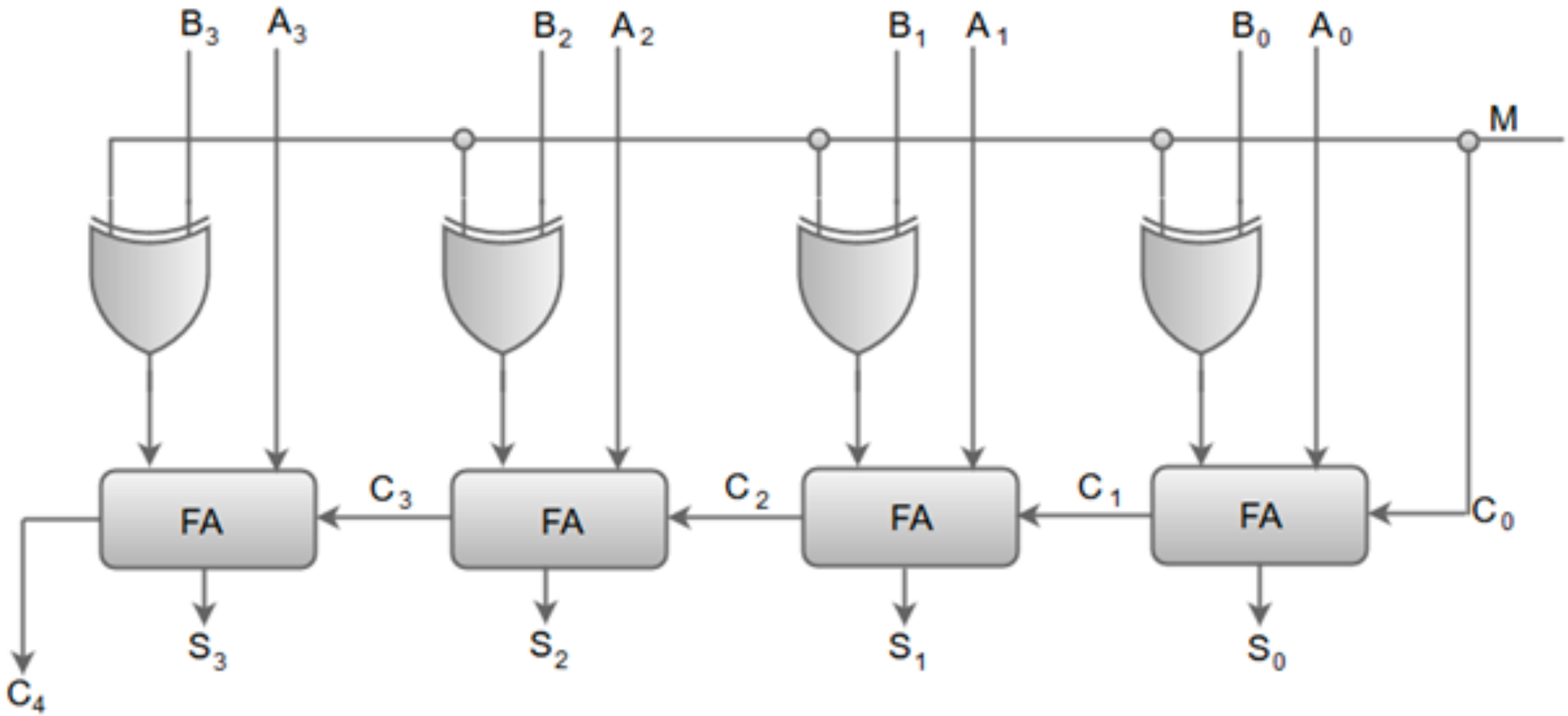


Figure 4-6. 4-bit binary adder

- The augend bits (A) and the addend bits (B) are designated by subscript numbers from right to left, with subscript '0' denoting the low-order bit.
- The carry inputs starts from C0 to C3 connected in a chain through the full-adders. C4 is the resultant output carry generated by the last full-adder circuit.
- The output carry from each full-adder is connected to the input carry of the next-high-order full-adder.
- The sum outputs (S0 to S3) generates the required arithmetic sum of augend and addend bits.
- The ***n*** data bits for the **A** and **B** inputs come from different source registers. For instance, data bits for **A** input comes from source register R1 and data bits for **B** input comes from source register R2.
- The arithmetic sum of the data inputs of A and B can be transferred to a third register or to one of the source registers (R1 or R2).

BINARY ADDER-SUBTRACTOR

4 bit adder-subtractor:



➢Mode input M controls the operation

- ➢ M=0 ---- adder
- ➢ M=1 ---- subtractor

- When the mode input (M) is at a low logic, i.e. '0', the circuit act as an adder and when the mode input is at a high logic, i.e. '1', the circuit act as a subtractor.
- The exclusive-OR gate connected in series receives input M and one of the inputs B.
- When M is at a low logic, we have $B \oplus 0 = B$.

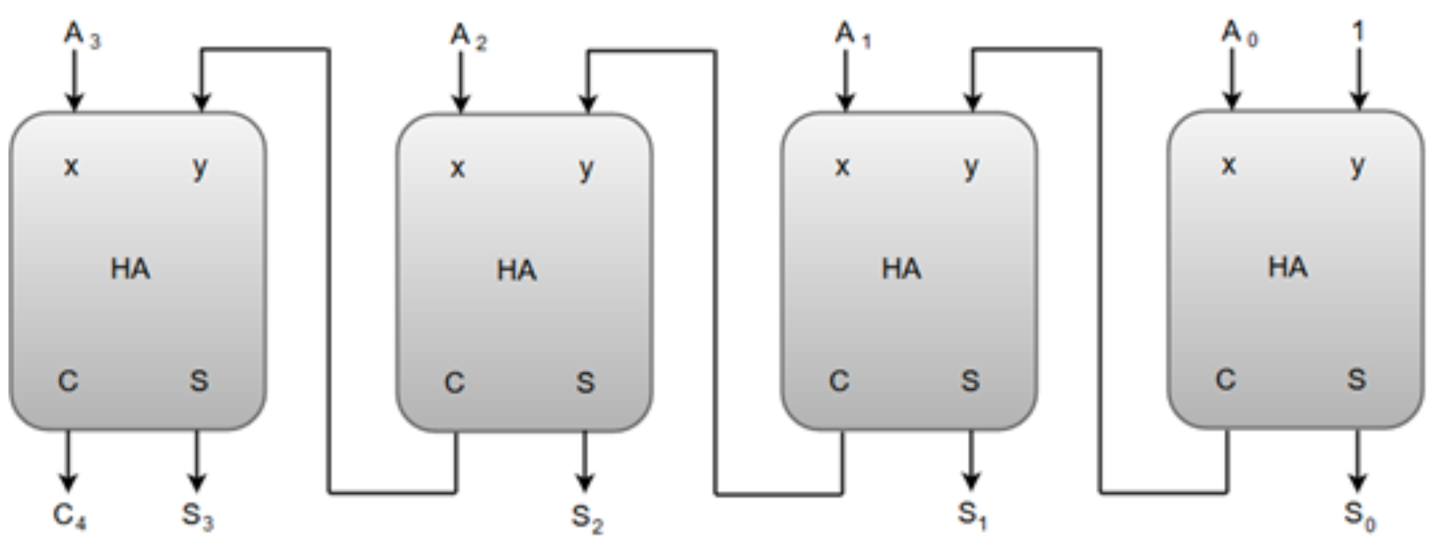
  The full-adders receive the value of B, the input carry is 0, and the circuit performs A plus B.
- When M is at a high logic, we have $B \oplus 1 = B'$ and $C0 = 1$.

  The B inputs are complemented, and a 1 is added through the input carry. The circuit performs the operation A plus the 2's complement of B.
-

BINARY INCREMENTER

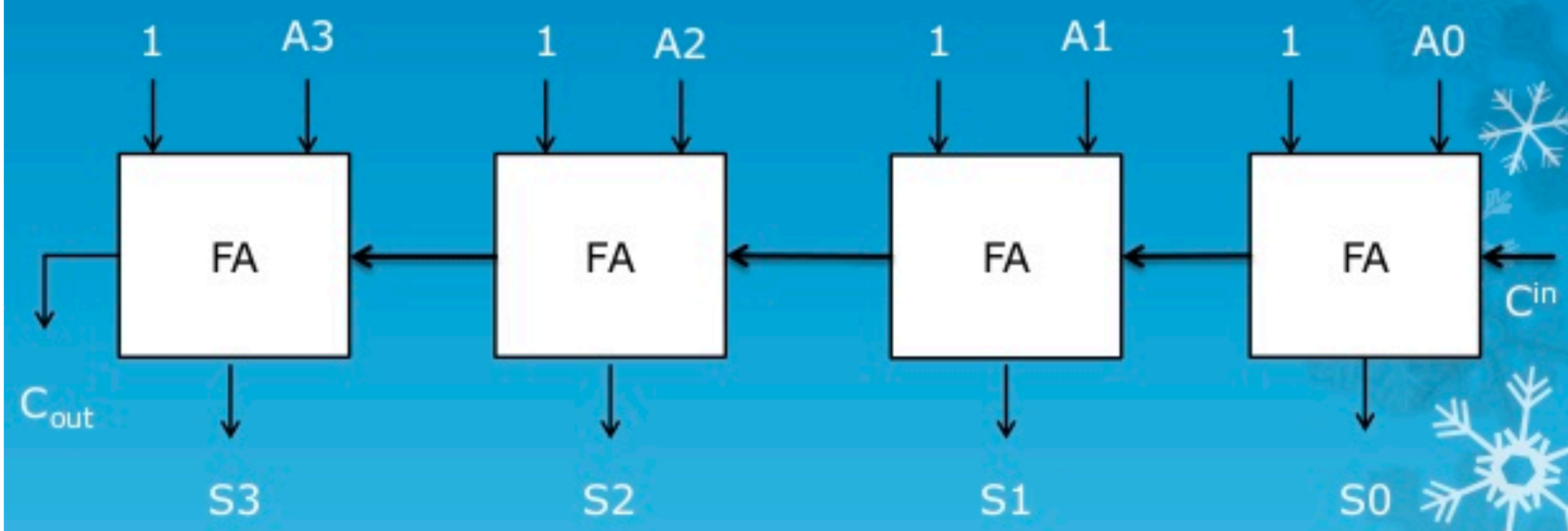
The increment micro-operation adds one binary value to the value of binary variables stored in a register. For instance, a 4-bit register has a binary value 0110, when incremented by one the value becomes 0111. The increment micro-operation is best implemented by a 4-bit combinational circuit incrementer. A 4-bit combinational circuit incrementer can be represented by the following block diagram.

4-bit binary incrementer:



- A logic-1 is applied to one of the inputs of least significant half-adder, and the other input is connected to the least significant bit of the number to be incremented.
- The output carry from one half-adder is connected to one of the inputs of the next-higher-order half-adder.
- The binary incrementer circuit receives the four bits from A0 through A3, adds one to it, and generates the incremented output in S0 through S3.
- The output carry C4 will be 1 only after incrementing binary 1111.

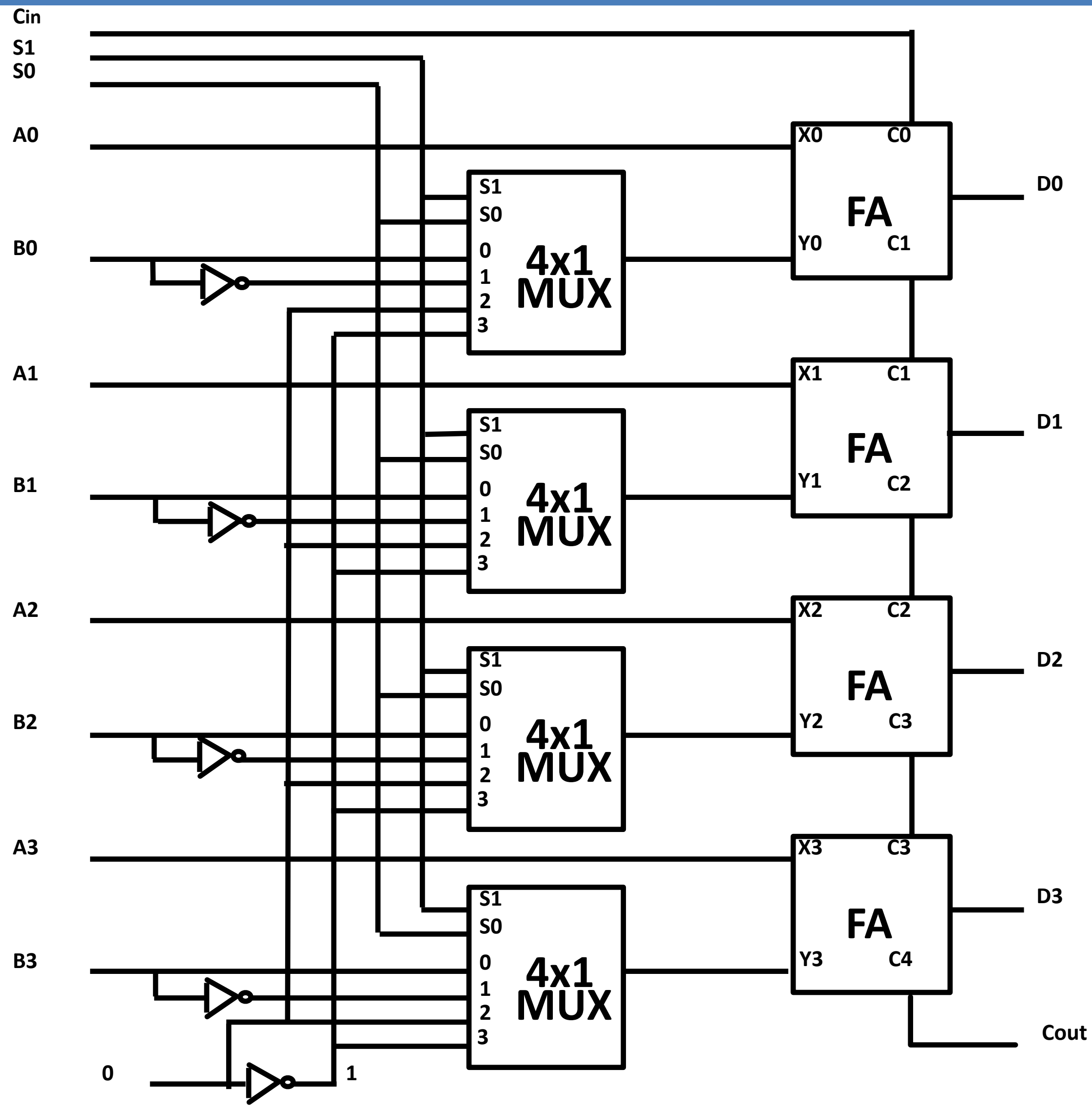
Binary Decrement Using Full Adder (4-bit)



BASIC THEORY

The binary decremented decreases the value stored in a register by '1'. For this, we can simply add '1' to the each bit of the existing value stored in a register. This is basically the concept of two's complement used for subtraction of '1' from given data. It is made by cascading 'n' full adders for 'n' number of bits i.e. the storage capacity of the register to be decremented. Hence, a 4-bit binary decrementer requires 4 cascaded half adder circuits. As stated above we add '1111' to 4 bit data in order to subtract '1' from it.

Arithmetic Circuits



Select			Input	Output
S1	S0	C_{in}	Y	$D=A+Y+C_{in}$
0	0	0	B	D=A+B
0	0	1	B	D=A+B+1
0	1	0	B'	D=A+B'
0	1	1	B'	D=A+B'+1
1	0	0	0	D=A
1	0	1	0	D=A+1
1	1	0	1	D=A - 1
1	1	1	1	D=A