Chapter-1
Register Transfer and Micro-operations

1.1 Register – Transfer Language
1.2 Register Transfer
1.3 Bus Transfer and Memory Transfer
1.4 Arithmetic Micro-Operations
1.5 Logic micro operations Shift Micro
1.6 operation. Arithmetic Logic Shift
1.7 Unit

Micro-operations

Ref. Book Name : Computer System Architecture, M. Morris Mano

- Micro-operations are elementary operations performed on data store in registers or in memory.

- The Micro-operations most frequently encountered are of four types:

  i. Transfer Micro-operations
  ii. Arithmetic Micro-operations
  iii. Logic Micro-operations
  iv. Shift Micro-operations

1.1 Register Transfer Language

Ref. Book Name : Computer System Architecture, M. Morris Mano

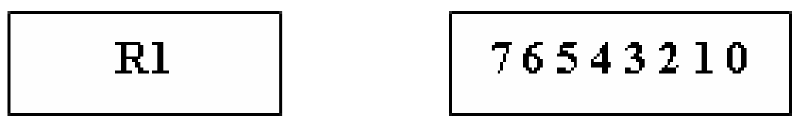
- The symbolic notation used to describe the micro-operation transfers among registers is called a register transfer language.
- Information transferred from one register to another is designated by means of a replacement operator.
- The statement Denotes a transfer of the content of register R1 into register R2.

$$R2 \leftarrow R1$$

- The content of source register R1 does not change after the transfer.

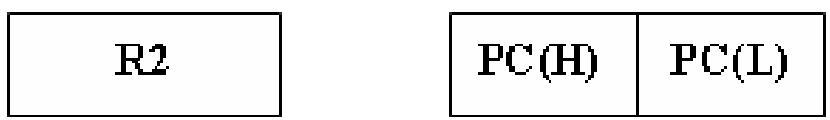
1.2 Register Transfer

- When data is transferred from one register to another register is known as register transfer.
- Computer registers are designated by capital letters to denote the function of the register.



(a) Register

(b) Showing individual bits



(c) Numbering of bits

(d) Divided into parts

1.2 Register Transfer

- To denote a transfer to occur only under a predetermined control condition

  if (P=1) then (R2 ← R1)

- Where P is a control signal generated in the control section.
- The control function is a Boolean variable that is equal to 1 or 0. It can also be denoted by

  P : R2 ← R1

- Control condition symbolizes the requirement that the transfer operation be executed by the hardware only if P=1.

Transfer between registers

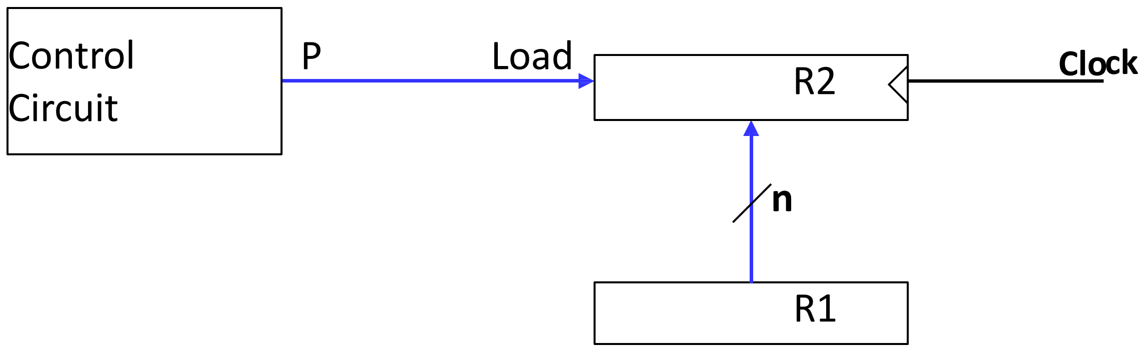
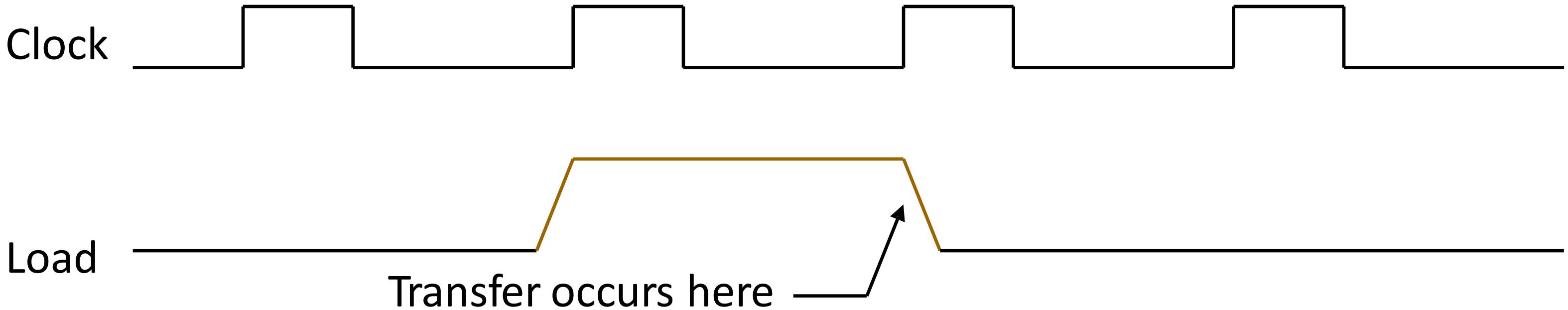


Figure: Transfer from R1 to R2, when P=1.



Timing diagram

1.3 Bus and Memory Transfers

- Transferring information between registers in a multiple register configuration is a common bus system.
- A bust structure consists of a set of common lines through which binary information is transferred one at a time.
- Control signals determine which register is selected by the bus during each particular register transfer.
- One way of constructing a common bus system is with multiplexers.
- The multiplexers select the source register whose binary information is then placed on the bus.
- The selection lines (S1, S0) choose the four bits of the register and transfer them to four line common bus.

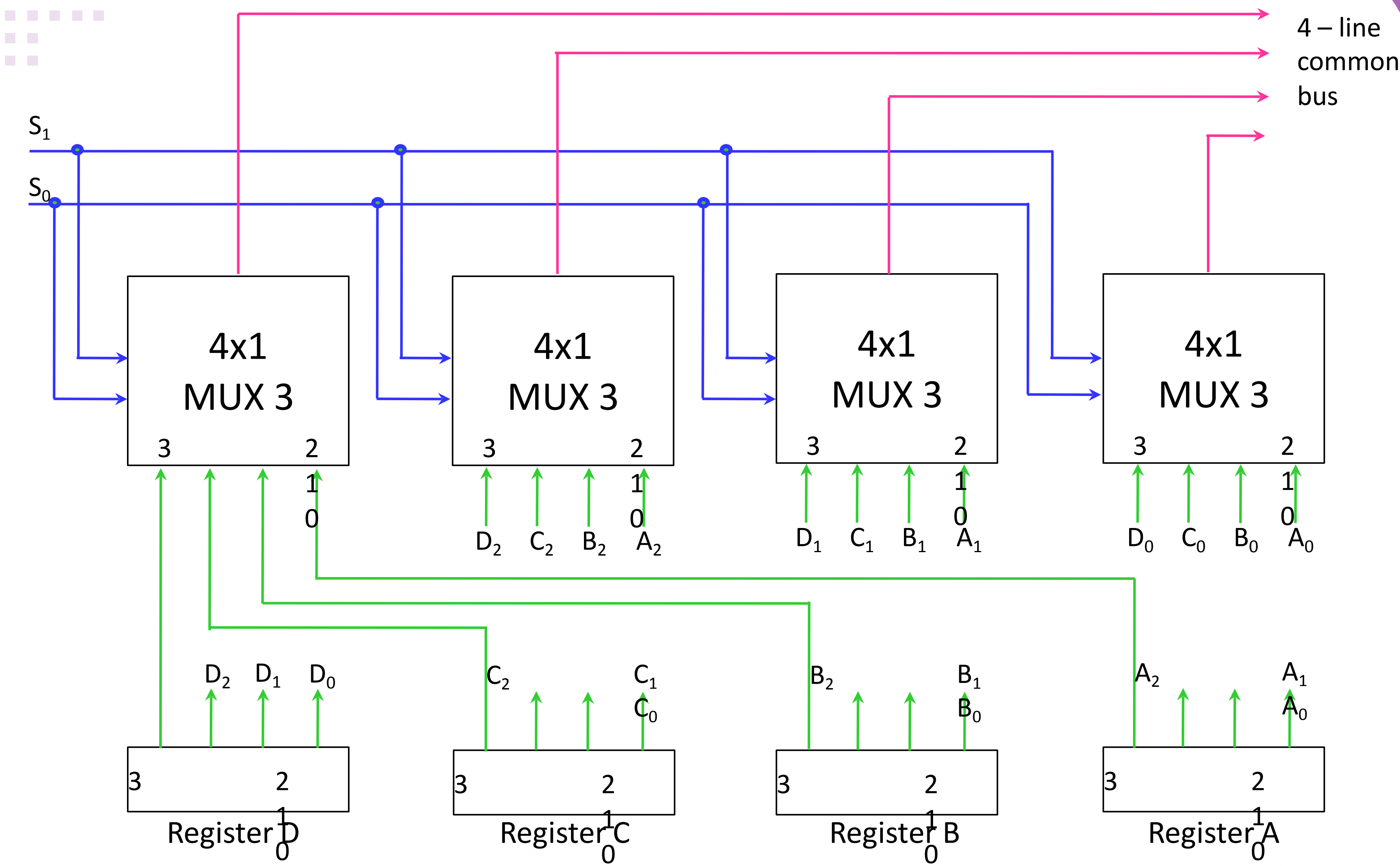
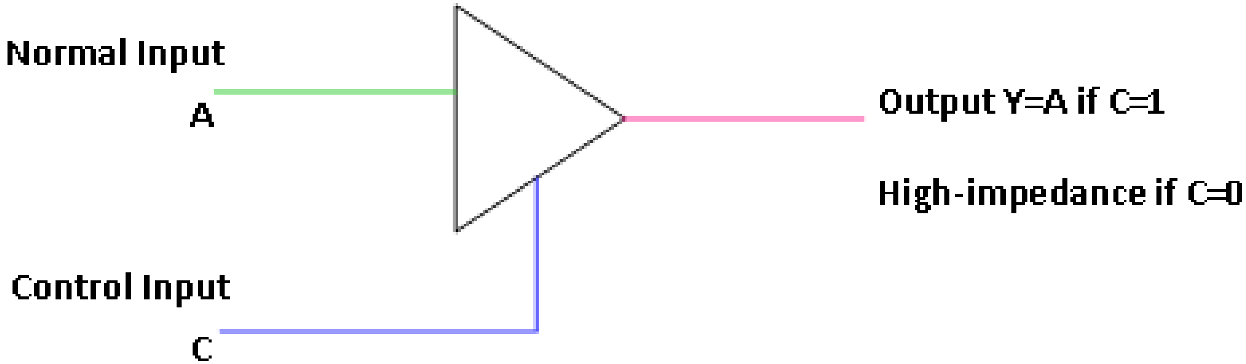


Figure: Bus system for four registers

Three state Bus Buffers

- Buffer is a circuit used to boost up the signals for transmitting over long distance.
- Three state gate is a digital circuit that exhibits three states.
- Two of the states are signals equivalent to logic 1 and 0.
- The third state is a high-impedance state.



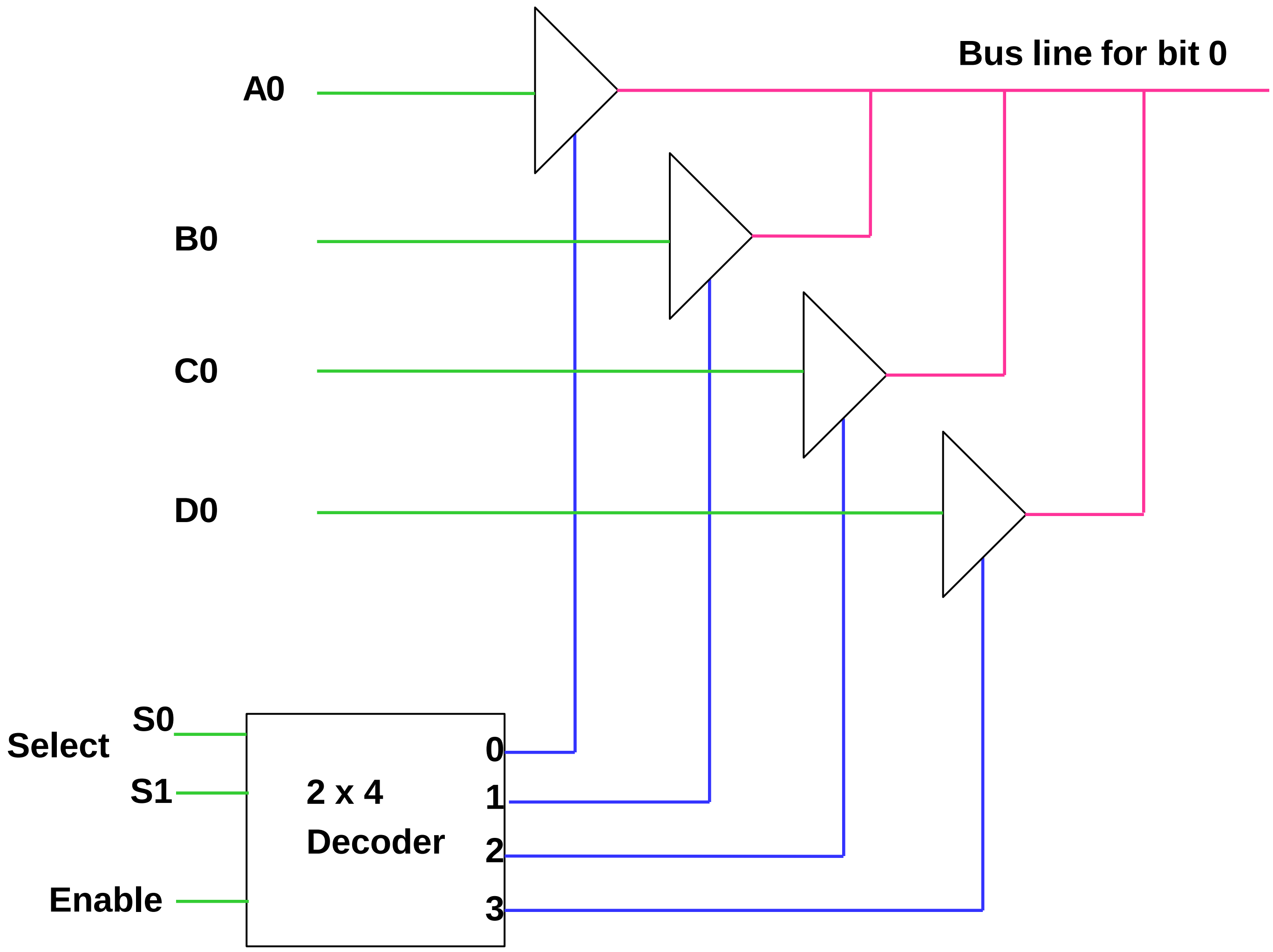
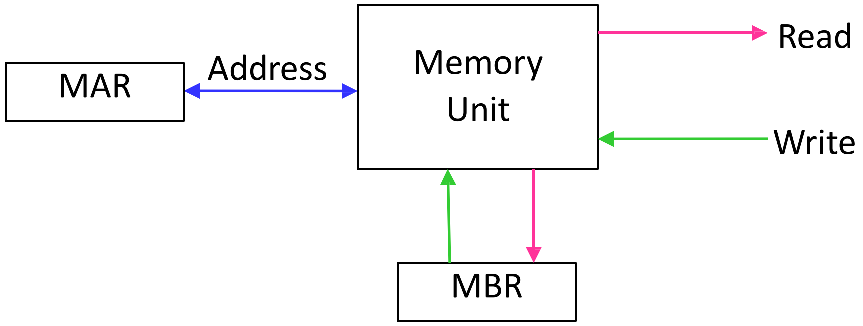


Figure: Bus Line with three state buffer

Memory Transfer

- The transfer of information from memory to external environment is called a read operation.
- Transfer of new information to be stored into memory is called a write operation.
- Read : DR ← M[AR]
- Write : M[AR] ← R1



R : MBR ← MAR

W : MAR ← MBR

[MAR : Memory Buffer Register]

[MBR : Memory Address Register]

1.4 Arithmetic Micro operations

Ref. Book Name : Computer System Architecture, M. Morris Mano

- Basic arithmetic micro operations are addition, subtraction, increment, decrement and shift.

- $R3 \leftarrow R1+R2$

- $R3 \leftarrow R1-R2$

- $R2 \leftarrow \overline{R2}$

- $R2 \leftarrow \overline{R2} +1$

Binary Adder

- Digital circuit that generates the arithmetic sum of two binary numbers of any length is called binary adder.

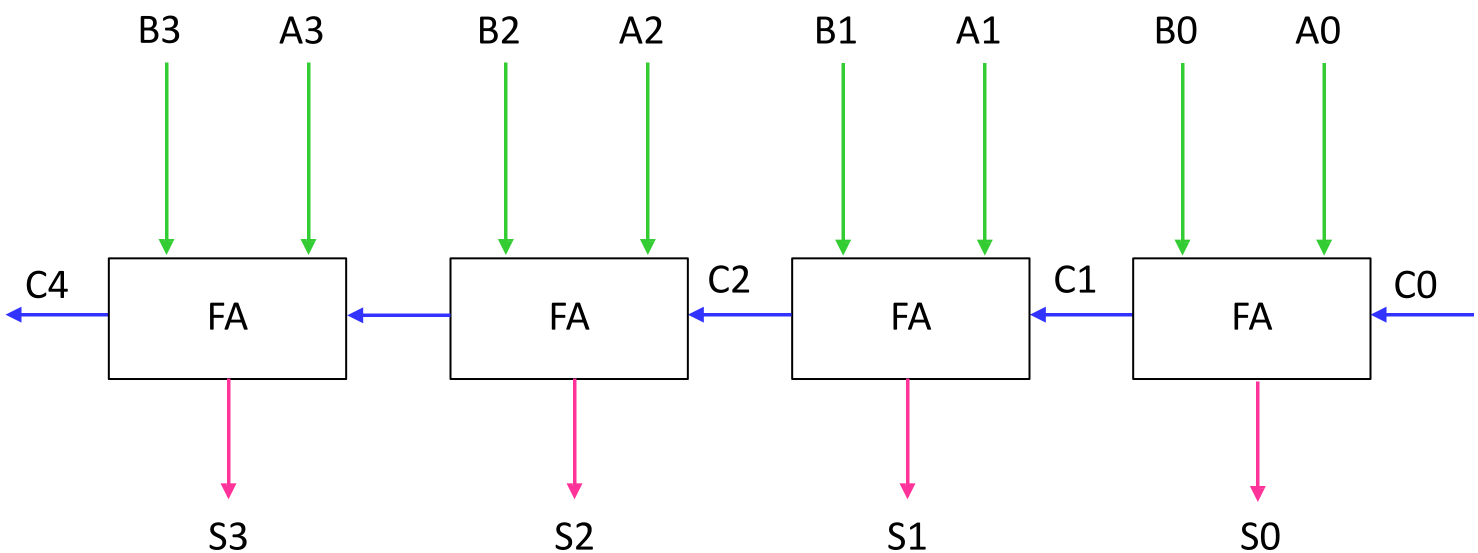


Figure: 4-bit binary adder

Binary Adder-Subtractor

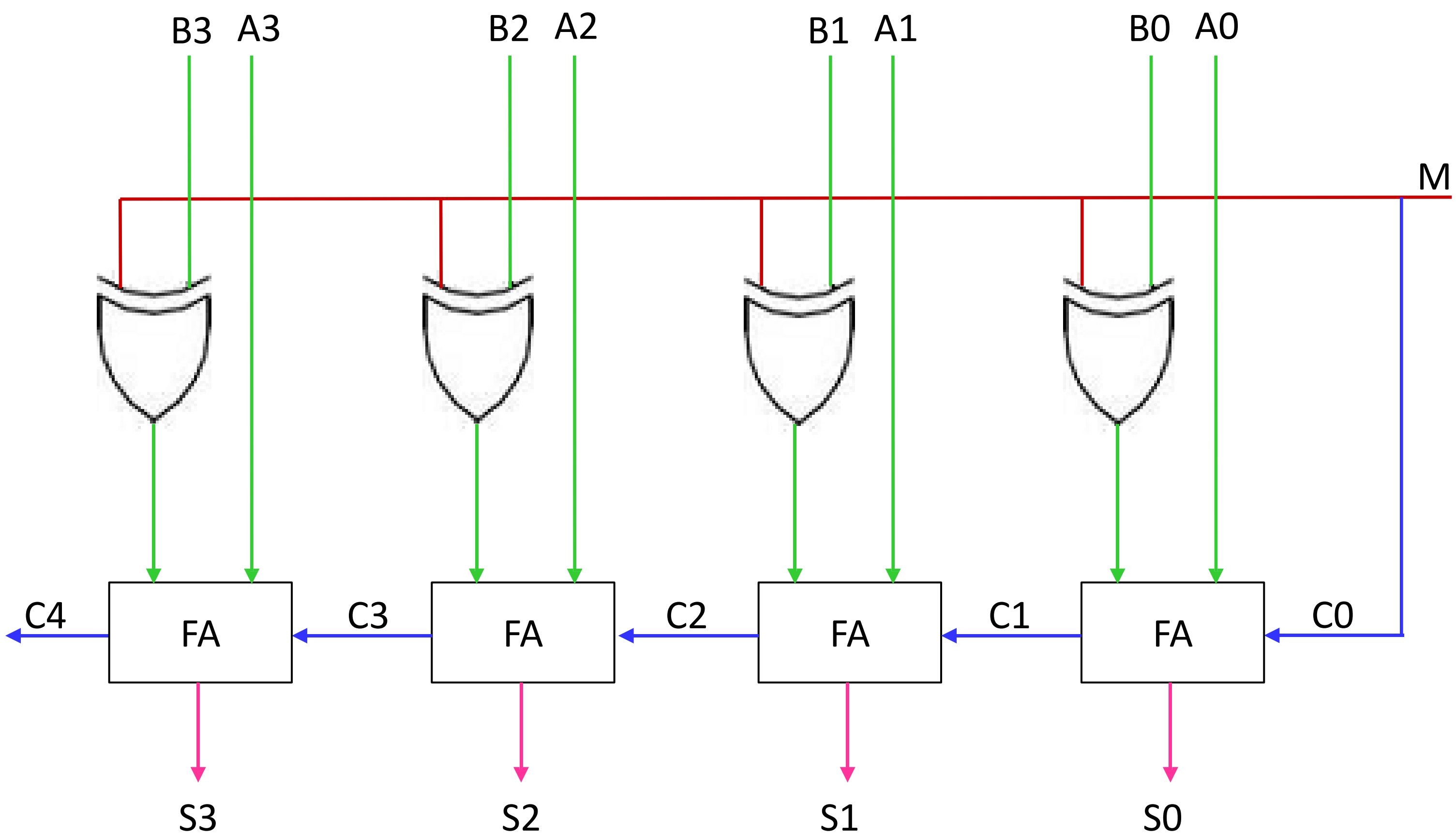


Figure: 4-bit binary adder-subtractor

Binary Adder-Subtractor

- The subtraction A-B can be done by taking the 2's complement of B and adding it to A.
- The mode input M controls the operation.
- When M=0 the circuit is an adder.
- When M=1 the circuit becomes a subtractor.
- Each exclusive-OR gate receives input M and one of the inputs of B.
- When M=0, we have $B \oplus 0 = B$, Full adders receive the value of B, input carry is 0 (C0 is connected to M), and circuit performs A+B.
- When M=1, we have $B \oplus 1 = B'$ and $C_0=1$. The B inputs are all complemented (B') and 1 is added through input carry.
- Circuit performs A + 2's complement of B.

Binary Adder-Subtractor

- Subtraction of B from A is written as
- A = A + 2’s complement of B =>(B’+1)
- For example,

  $A = 1011_2 = 11_{10}$

  $B = 1001_2 = 9_{10}$
- 1’s complement of register B is $B' = 0110_2$
- 2’s complement of B is $B' + 1 = 0110 + 1 = 0111_2$
- Now add A to 2’s complement of B

```
    1011  ------------>  A
  + 0111  ------------>  2’s complement of B
  -----------
  1)0010   =   2₁₀
```

Output carry

Result

Binary Incrementer

- The increment micro-operation adds one to a number in register.

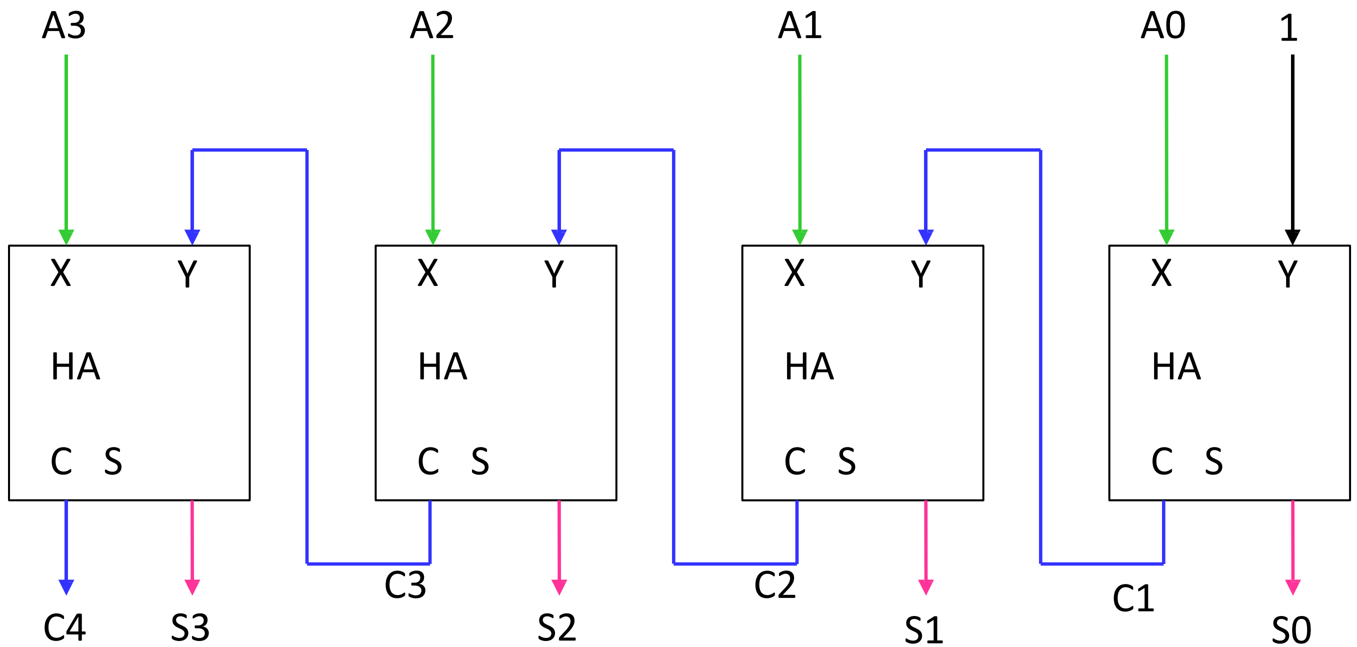


Figure: 4-bit binary incrementer

Binary Incrementer

- One of the inputs of the half-adder is connected to logic 1.
- Other input is connected to the least significant bit of the number to be incremented.
- Output carry from one half-adder is connected to one of the inputs of the next higher order half-adder.
- Circuit receives four bits from A0 through A3, adds 1 to it, and generates the incremented output S0 through S3.
- Output carry C4 will be 1 only after incrementing binary 1111.
- Circuit can be extended to an n-bit binary incrementer by extending the diagram to include n half-adders.

Arithmetic Circuit

- The basic component of an arithmetic circuit is the parallel adder.
- The four inputs from A go directly to the X inputs of the binary adder.
- Each of the four inputs from B are connected to the data inputs of the multiplexers.
- The data input of multiplexers also receive complement of B.
- The other two data inputs are connected to logic-0 and 1.
- The four multiplexers are controlled by two selection inputs, S1 and S0.
- The output of the binary adder is calculated from the following arithmetic sum:

  $D = A + Y + C_{in}$

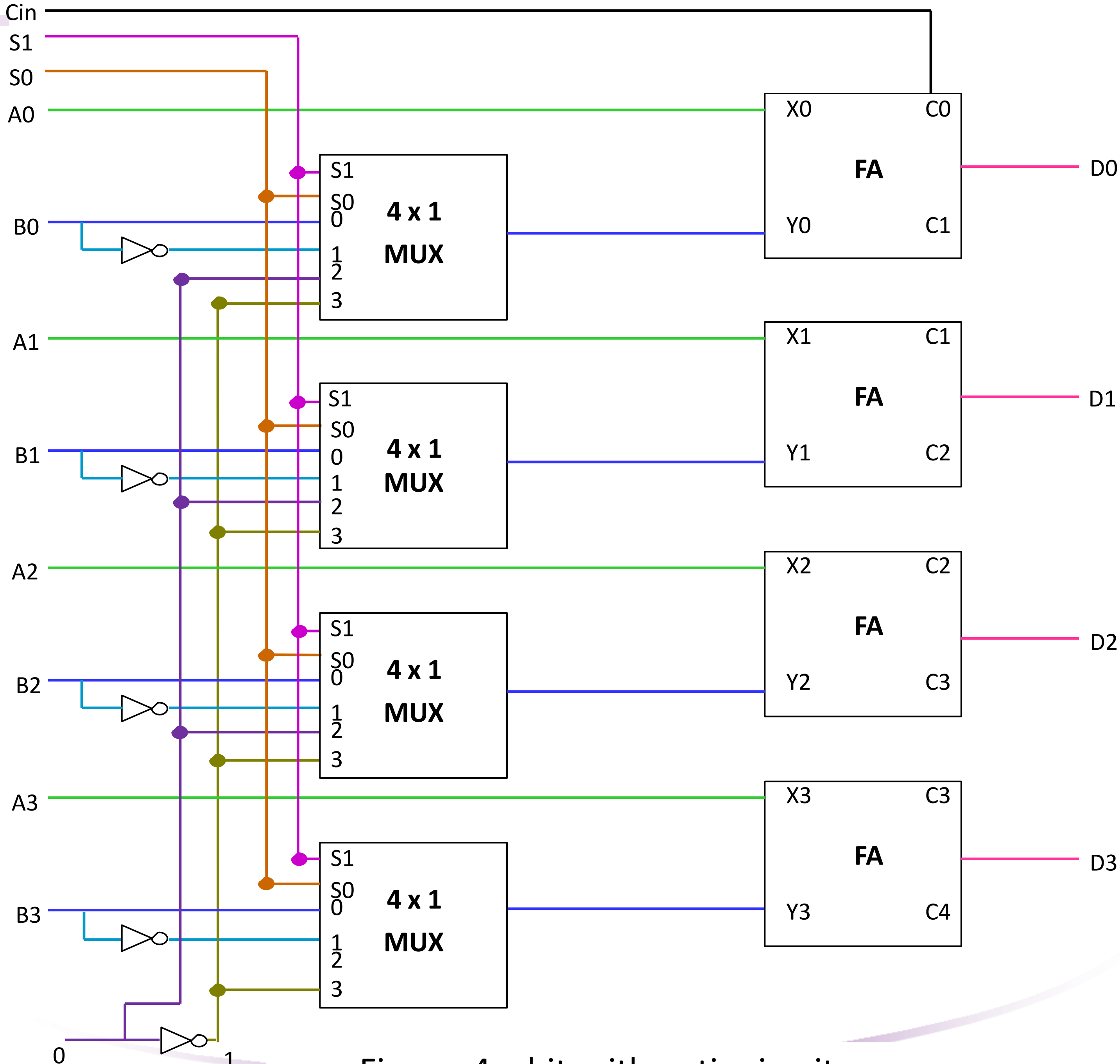


Figure: 4 – bit arithmetic circuit

Arithmetic circuit function table

Select			Input	Output	Micro-operation
S1	S0	Cin	Y	D=A+Y+Cin	
0	0	0	B	D=A+B	Add
0	0	1	B	D=A+B+1	Add with carry
0	1	0	B’	D=A+B’	Subtract with borrow
0	1	1	B’	D=A+B’+1	Subtract
1	0	0	0	D=A	Transfer A
1	0	1	0	D=A+1	Increment A
1	1	0	1	D=A-1	Decrement A
1	1	1	1	D=A	Transfer A

1.5 Logic Micro-operations

Ref. Book Name : Computer System Architecture, M. Morris Mano

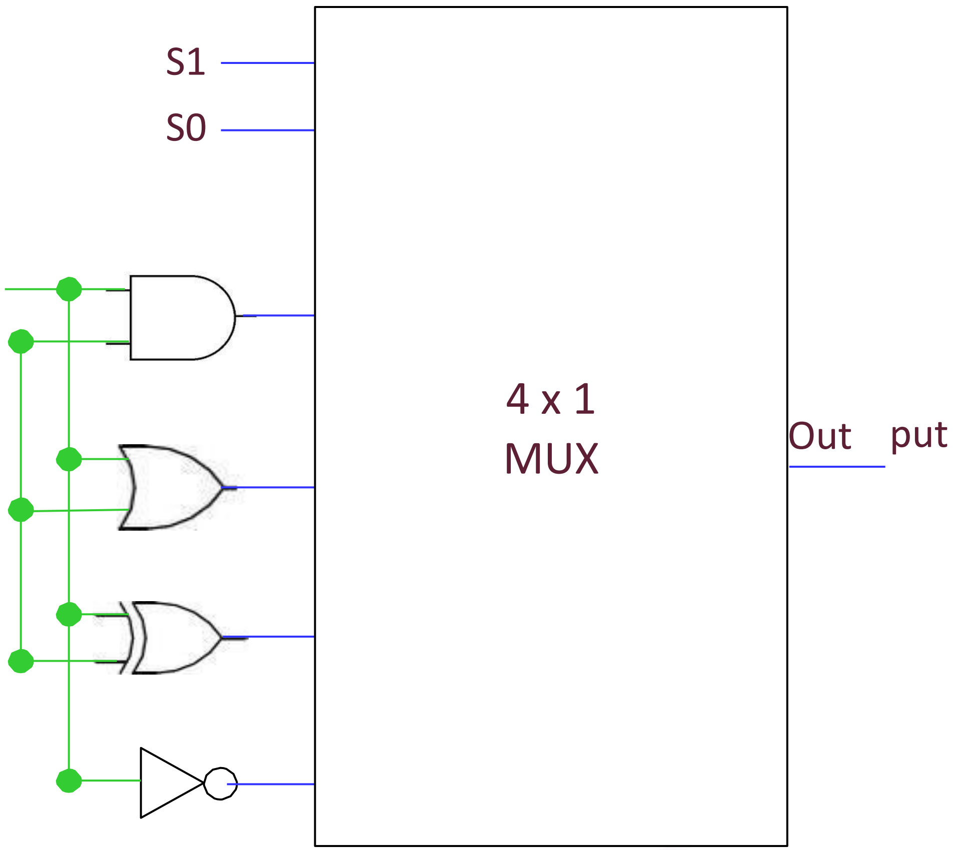
X	**0**	**0**	**1**	**1**	**Boolean Function**	**Micro-operation**	**Name**
Y	**0**	**1**	**0**	**1**			
	0	0	0	0	F0 = 0	$F \leftarrow 0$	Clear
	0	0	0	1	F1 = XY	$F \leftarrow A \wedge B$	AND
	0	0	1	0	F2 = XY’	$F \leftarrow \overline{A} \wedge B$	
	0	0	1	1	F3 = X	$F \leftarrow A$	Transfer A
	0	1	0	0	F4 = X’Y	$\overline{F} \leftarrow A \wedge B$	
	0	1	0	1	F5 = Y	$F \leftarrow B$	Transfer B
	0	1	1	0	F6 $\oplus$ Y	F $\oplus$ A B	Exclusive OR
	0	1	1	1	F7 = X + Y	$F \leftarrow A \vee B$	OR

1.5 Logic Micro-operations

X	**0**	**0**	**1**	**1**	**Boolean Function**	**Micro-operation**	**Name**
Y	**0**	**1**	**0**	**1**			
	1	0	0	0	F8 = (X + Y)’	$F \leftarrow \overline{A \vee B}$	NOR
	1	0	0	1	F9 = (X $\oplus$ Y)’	$F \leftarrow \overline{A \oplus B}$	Exclusive NOR
	1	0	1	0	F10 = Y’	$\overline{F} \leftarrow B$	Complement B
	1	0	1	1	F11 = X + Y’	$F \leftarrow A \vee \overline{B}$	
	1	1	0	0	F12 = X’	$\overline{F} \leftarrow A$	Complement A
	1	1	0	1	F13 = X’ + Y	$F \leftarrow \overline{A} \vee B$	
	1	1	1	0	F14 = (XY)’	$F \leftarrow \overline{A \wedge B}$	NAND
	1	1	1	1	F15 = 1	$F \leftarrow$ all 1’s	Set to all 1’s

Hardware Implementation

- Logic micro-operations specify binary operations for strings of bits stored in register.
- Most computers use only 4 micro-operations :
  1. AND
  2. OR
  3. XOR
  4. Complement



S1	S0	Output	Operation
0	0	$E = A \wedge B$	AND
0	1	$E = A \vee B$	OR
1	0	$E = A \oplus B$	XOR
1	1	$E = \overline{A}$	Complement

Figure: One stage of logic circuit

Applications of Logic Micro-operation

- **Selective set**
  - The selective set operation sets to 1 the bits in register A where there are corresponding 1's in register B.
  - It does not affect bit positions that have 0's in B.
  - For example:

1 0 1 0	A before
1 1 0 0	B logic operand
1 1 1 0	A after

  - A: Processor Register
  - B: Logic operand extracted from memory
  - OR micro-operation can be used to selectively set bits of a register.

Selective Complement

- This operation complements bits in A where there are corresponding bits in A where there are corresponding 1’s in B.
- For example,

1 0 1 0	A before
1	B logic operand
1 0 0 0 0 1 1 0	A after

- Exclusive-OR micro-operation can be used for selective complement.

Selective Clear

- The selective clear operation clears to 0 the bits in A only where there are corresponding 1’s in B.
- For example,

1 0 1 0	A before
1	B logic operand
1 0 0 0 0 1 1 0	A after

- Corresponding logic micro-operation is A ← A ∧ B

Mask (Delete)

- The mask operation is similar to the selective clear operation except that the bits of A are cleared only where there are corresponding 0's in B.

- For example,

1 0 1 0	A before
1 1 0 0	B logic operand
1 0 0 0	A after

- The mask operation is an AND micro-operation

Insert

- The insert operation inserts a new value into group of bits.

- This is done by first masking the bits and then ORing them with the required value.
- For example,

0 1 1 0 1 0 1 0	A before
0 0 0 0 1 1 1 1	B (mask)
0 0 0 0 1 0 1 0	A after masking

- Now insert the new value

0 1 1 0 1 0 1 0	A before
1 0 0 1 1 1 1 1	B (insert)
1 0 0 1 1 0 1 0	A after insertion

1.6 Shift Micro-operations

Ref. Book Name : Computer System Architecture, M. Morris Mano

- Shift micro-operations are used for serial transfer of data.
- The contents of a register can be shifted to the left or to the right.
- The information transferred through the serial input determines the type of shift. There are three types of shifts:

1. Logical Shift
2. Circular Shift
3. Arithmetic Shift

1.6 Shift Micro-operations

1. Logical Shift

- A logical shift is one that transfers 0 through the serial input.

- The symbols shl and shr denotes logical shift.

R1 $\leftarrow$ shl R1

R2 $\leftarrow$ shr R2.

2. Circular Shift (rotate operation)

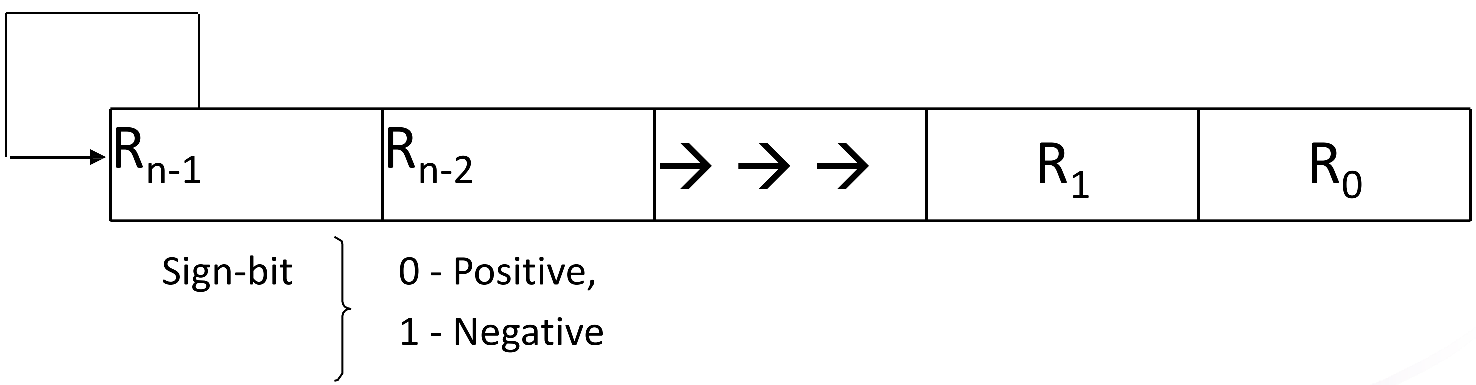
- The circular shift circulates the bits of the register around the two ends without loss of information.

- cil denotes circular shift left
- cir denotes circular shift right.

1.6 Shift Micro-operations

3. Arithmetic Shift

- An arithmetic shift is a micro-operation that shifts a signed binary number to the left or right.
- An arithmetic shift-left multiplies a signed binary number by 2.
- An arithmetic shift-right divides the number by 2.



1.7 Arithmetic Logic Shift Unit

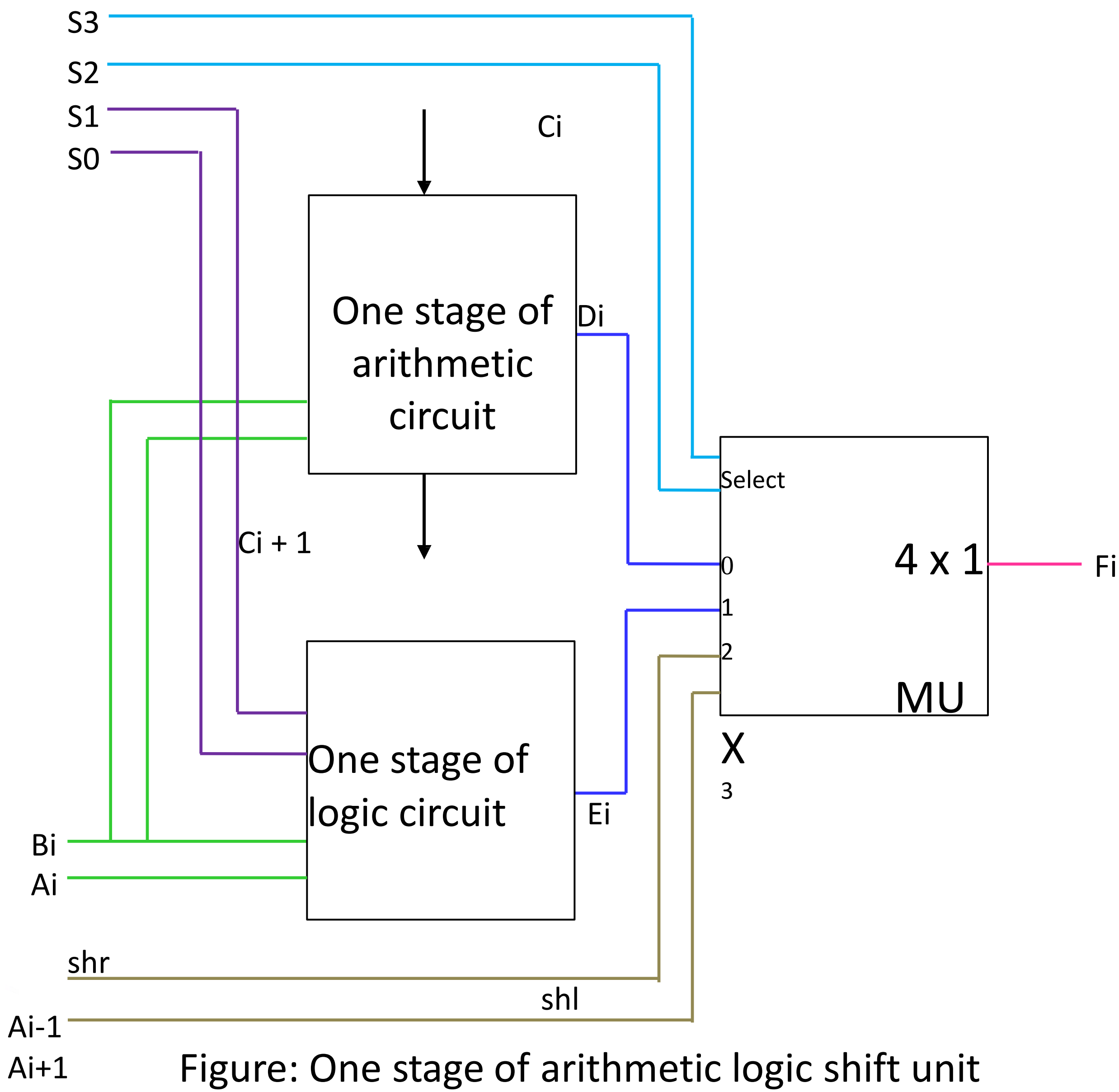


Figure: One stage of arithmetic logic shift unit

Function table for Arithmetic Logic Shift Unit

Operation Select				Cin	Operation	Function
S3	S2	S1	S0			
0	0	0	0	0	F = A	Transfer A
0	0	0	0	1	F = A + 1	Increment A
0	0	0	1	0	F = A + B	Addition
0	0	0	1	1	F = A + B + 1	Add with carry
0	0	1	0	0	F = A + B’	Subtract with borrow
0	0	1	0	1	F = A + B’ + 1	Subtraction
0	0	1	1	0	F = A – 1	Decrement
0	0	1	1	1	F = A	Transfer A
0	1	0	0	X	$F = A \wedge B$	AND
0	1	0	1	X	$F = A \vee B$	OR
0	1	1	0	X	$F = A \oplus B$	XOR
0	1	1	1	X	F = A’	Complement
1	0	X	X	X	F = shr A	Shift right A into F
1	1	X	X	X	F = shl A	Shift left A into F