Bus and Memory Transfer

BUS

- A bus is a collection of wires through which data is transmitted from one part of a computer to another.

- The size of a bus, known as its ***width,*** it determines how much data can be transmitted at one time. For example, a 16-bit bus can transmit 16 bits of data, whereas a 32-bitbus can transmit 32 bits of data.

Types of Computer Bus

Data Bus: The data bus allows data to travel back and forth between the microprocessor (CPU) and memory (RAM).

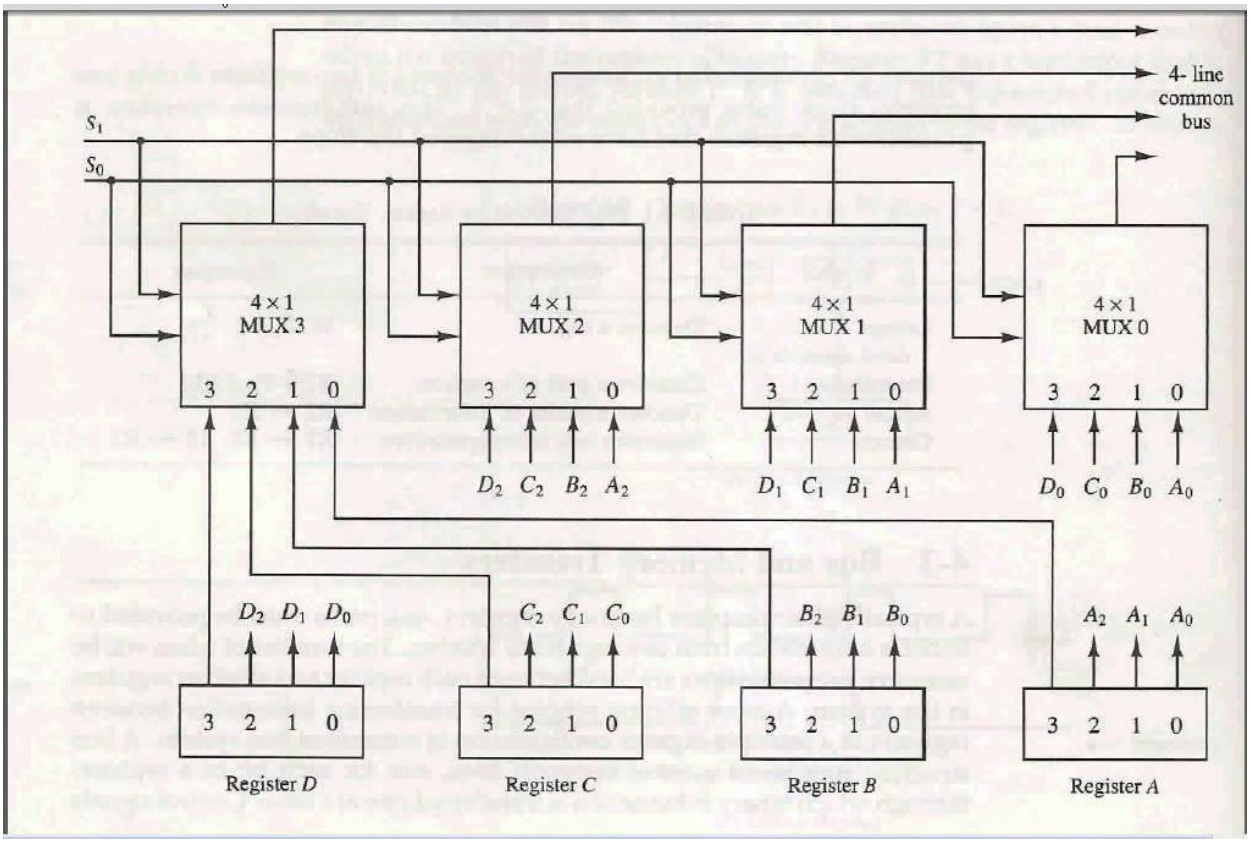
Address Bus: The address bus carries information about the location of data in memory.

Control Bus: The control bus carries the control signals that make sure everything is flowing smoothly from place to place.

Memory 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 construction of a bus system for four registers is shown in diagram.



- Each register has four bits, numbered 0 through 3. The bus consists of four 4 ×1 multiplexers each having four data inputs, 0 through 3, and two selection inputs, S1and S0.

- Thus MUX 0 multiplexes the four 0 bits of the registers, MUX 1 multiplexes the four 1 bits of the registers, and similarly for the other two bits.

- The two selection lines S1and S0 are connected to the selection inputs of all four multiplexers.

Truth Table

S_1	S_0	Register selected
0	0	*A*
0	1	*B*
1	0	*C*
1	1	*D*

- The selection lines choose the four bits of one register and transfer them into the four-line common bus. When S1S0= 00, the 0 data inputs of all four multiplexers are selected and applied to the outputs that form the bus.

- This causes the bus lines to receive the content of register A since the outputs of this register are

connected to the 0 data inputs of the multiplexers. Similarly, register B is selected if S1S0= 01, and so on.

- The transfer of information from a bus into one of many destination registers can be accomplished by connecting the bus lines to the inputs of all destination registers and activating the load control of the particular destination register selected.

- The symbolic statement for a bus transfer may mention the bus or its presence may be implied in the statement. When the bus is includes in the statement, the register transfer is symbolized as follows:

  BUS ←C, R1 ←BUS

- The content of register C is placed on the bus, and the content of the bus is loaded into register R1 by activating its load control input. If the bus is known to exist in the system, it may be convenient just to show the direct transfer.

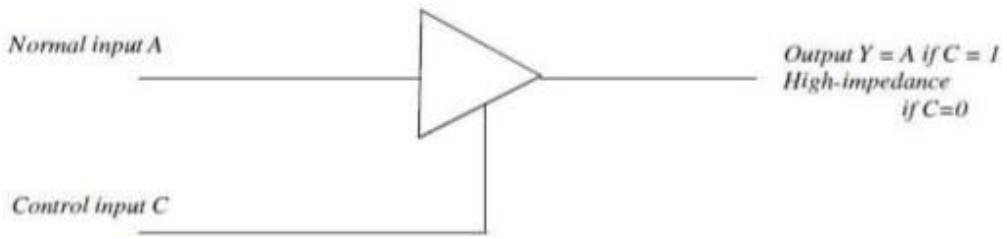
  R1 ←C

THREE-STATE BUS BUFFERS

- A three-state gate is a digital circuit that exhibits three states. Two of the states are signals equivalent to logic 1 and 0 as in a conventional gate. **The third state is a high-impedance state.**

- **The high-impedance state (Disable) behaves** like an open circuit which means that the output is disconnected and does not have logic significance.

- The graphic symbol of a three-state buffer gate is shown below.



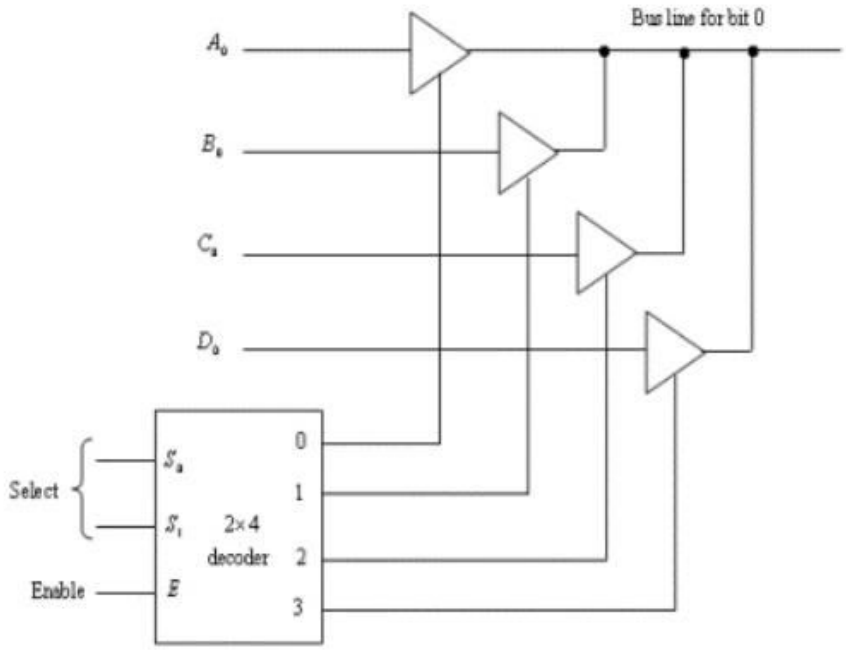
- It is distinguished from a normal buffer by having both a normal input and a control input. The control input determines the output state.

- **When the control input is equal to 1**, the output is enabled and the gate behaves like any conventional buffer, with the output equal to the normal input.

- **When the control input is 0,** the output is disabled and the gate gives to a high-impedance state.

- The construction of a bus system with three-state buffers is shown in following diagram.

BUS LINE WITH THREE STATE BUFFER



- The outputs of four buffers are connected together to form a single bus line.

- **The control inputs to the buffers** determine which of the four normal inputs will communicate with the bus line.

- No more than one buffer may be in the active state at any given time. The connected buffers must be controlled so that only one three-state buffer has access to the bus line while all other buffers are maintained in a high-impedance state. **So for this purpose decoder is used.**

- **When the enable input of the decoder is 0**, all of bits four outputs are 0, and the bus line is in a high-impedance state because all four buffers are disabled.

- **When the enable input is active**, one of the three-state buffers will be active, depending on the binary value in the select inputs of the decoder.

MEMORY TRANSFER

- The memory unit supports two fundamental operations: Read and Write.

- The read operation read a previously stored data, executed by the processor to read a data byte from memory.

- The write operation stores a value in memory, executed by processor to write a data byte in a memory location.

- Consider a memory unit that receives the address form a register, called the address register, symbolized by AR. The data are transferred to another register, called the data register, symbolized by DR. the read operation can be stated as follows:

  Read: DR ←M [AR]

- Assume that the input data are in register R1 and the address is in AR. The write operation can be stated symbolically as follows:

  Write: M [AR] ←R1

ARITHMETIC MICRO OPERATIONS

- The micro operation is inside the computer and for executing any instruction CPU has to perform some micro operation.

- A micro operation is an elementary operation performed with the data stored in registers.

- The micro operations most often encountered in digital computers are classified into **four categories**:

  1. **Register transfer micro operations** transfer binary information from one register to another.

  2. **Arithmetic micro operations** perform arithmetic operation on numeric data stored in registers.

  3. **Logic micro operations** perform bit manipulation operations on non-numeric data stored in registers.

  4. **Shift micro operations** perform shift operations on data stored in registers.

- The register transfer micro operation does not change the information content when the binary information moves from the source register to the destination register.

- The other three types of micro operation change the information content during the transfer. In this section we introduce a set of arithmetic micro operations.

Arithmetic micro operations

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

- Arithmetic shifts are explained later in conjunction with the shift micro operations. The arithmetic micro operation defined by the statement specifies an add micro operation.

$$R3 \leftarrow R1 + R2$$

- It states that the contents of register R1 are added to the contents of register R2 and the sum transferred to register R3.

- **Subtraction** is most often implemented through complementation and addition. Instead of using the minus operator, we can specify the subtraction by the following statement:

$$R3 \leftarrow R1 + \overline{R2} + 1$$

- $\overline{R2}$ is the symbol for the 1's complement of R2. Adding 1 to the 1's complement produces the 2's complement. Adding the contents of R1 to the 2's complement of R2 is equivalent to R1 – R2.

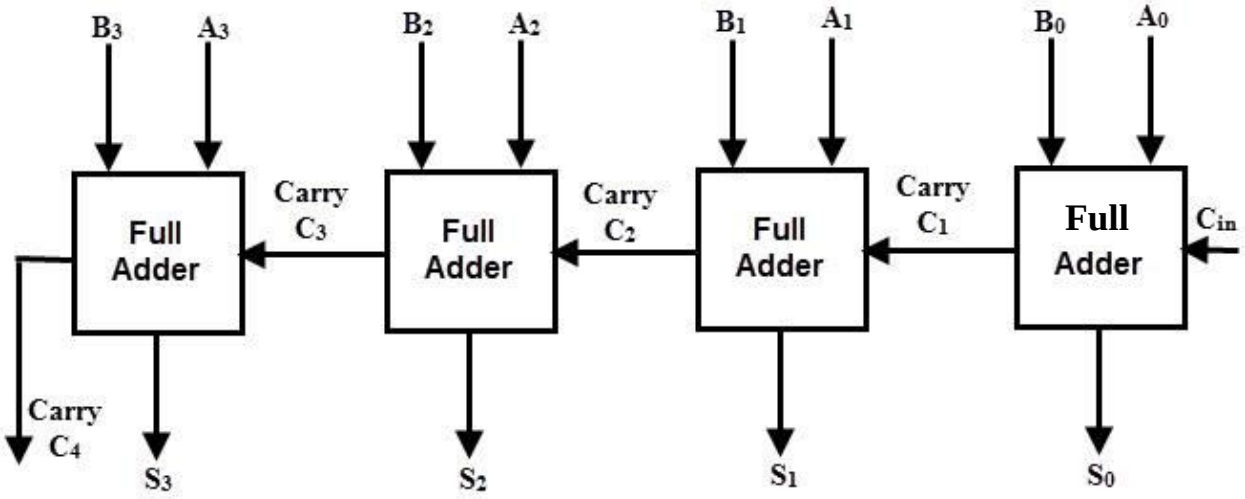
- **The multiplication operation** is implemented with a sequence of add and shift micro operations. **Division** is implemented with a sequence of subtract and shift micro operations.

TABLE Arithmetic Microoperations

Symbolic designation	Description
$R3 \leftarrow R1 + R2$	Contents of R1 plus R2 transferred to R3
$R3 \leftarrow R1 - R2$	Contents of R1 minus R2 transferred to R3
$R2 \leftarrow \overline{R2}$	Complement the contents of R2 (1's complement)
$R2 \leftarrow \overline{R2} + 1$	2's complement the contents of R2 (negate)
$R3 \leftarrow R1 + \overline{R2} + 1$	R1 plus the 2's complement of R2 (subtraction)
$R1 \leftarrow R1 + 1$	Increment the contents of R1 by one
$R1 \leftarrow R1 - 1$	Decrement the contents of R1 by one

BINARY ADDER

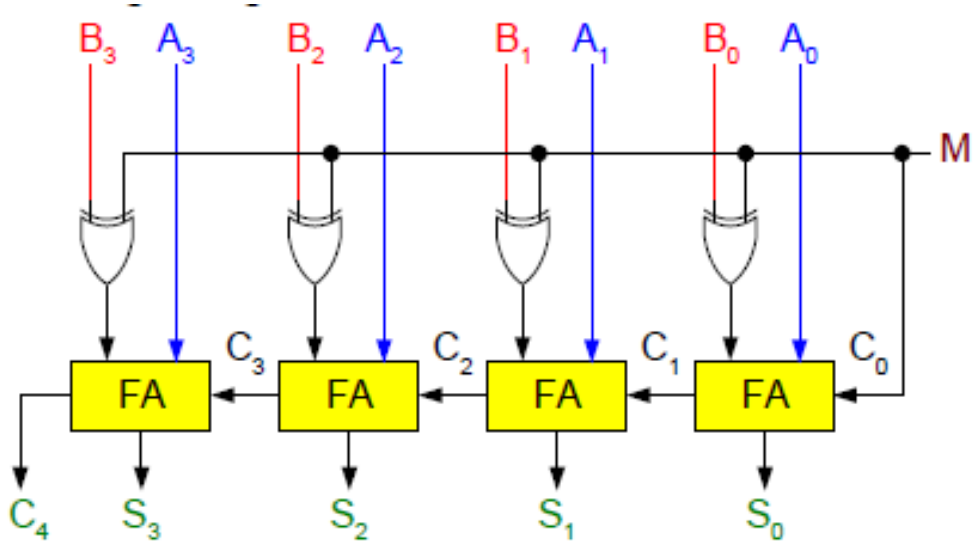
- The digital circuit that generates the arithmetic sum of two binary numbers of any lengths is called a binary adder. The binary adder is constructed with full-adder circuits connected in cascade.



- The output carry from one full-adder connected to the input carry of the next full-adder.
- The bits of A and the bits of B are designated by subscript numbers from right to left.(from registers)
- With subscript 0 denoting the low-order bit. The carries are connected in a chain through the full-adders. The input carry to the binary adder is C0 and the output carry is C4.
- The S outputs of the full-adders generate the required sum bits.

Binary Adder-Subtractor

- The addition and subtraction operations can be combined into one common circuit by including an exclusive-OR gate with each full-adder.



- **The mode input M controls the operation**. **When M = 0** the circuit is an adder and **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. the full-adders receive the value of B, the input carry is 0, and the circuit performs A plus B. **When M = 1**, we have B $\oplus$1 = B' and C0=1.

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

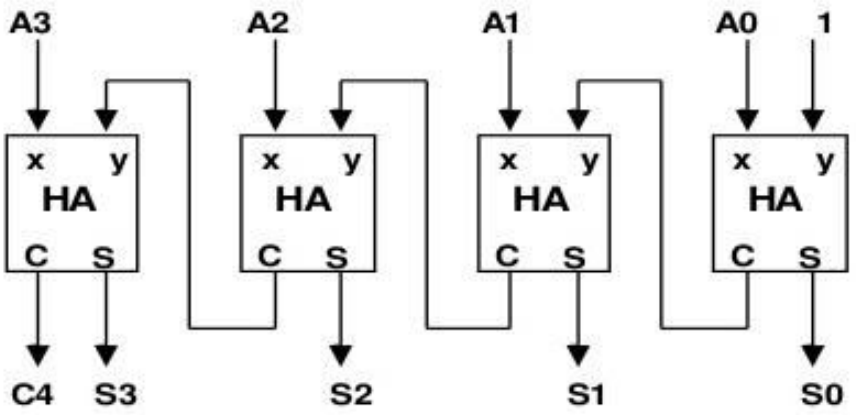
BINARY INCREMENTER

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

- **For example**, if a 4-bit register has a binary value 0110, it will go to 0111 after it is incremented.

- This micro operation is easily implemented with a binary counter. Every time the count enable is active, the clock pulse transition increments the content of the register by one.

- This can be accomplished by means of half-adders connected in cascade.



- The least significant bit must have one input connected to logic-. The other inputs receive the number to be incremented or the carry from the previous stage.

- The output carry from one half-adder is connected to one of the inputs of the next-higher-order half-adder. The circuit receives the four bits from A0through A3, adds one to it, and generates the incremented output in S0through S3.

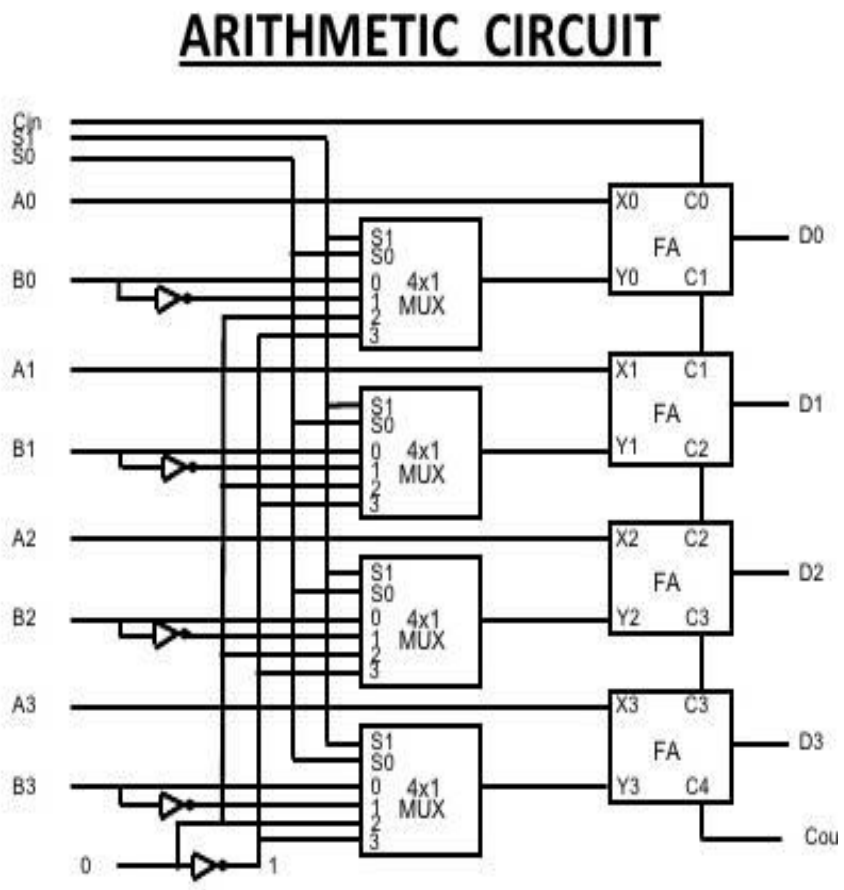
- The output carry C4will be 1 only after incrementing binary 1111. This also causes outputs S0 through S3 to go to 0.

ARITHMETIC CIRCUIT

- The arithmetic micro operations can be implemented in one composite arithmetic circuit.

- The basic component of an arithmetic circuit is the parallel adder. By controlling the data inputs to the adder, it is possible to obtain different types of arithmetic operations.

- The diagram of a 4-bit arithmetic circuit is shown below



- It has four full-adder circuits that constitute the 4-bit adder and four multiplexers for choosing different operations. There are two 4-bit inputs A and B and a 4-bit output D.

- The four inputs from A go directly to the X inputs of the binary adder. Each of the for inputs from B are connected to the data inputs of the multiplexers. The multiplexer's data inputs also receive the complement of B.

- The other two data inputs are connected to logic-0 ad logic -1. Logic-0 is fixed voltage value (0 volts for TTL integrated

circuits) and thelogic-1 signal can be generated through an inverter whose input is 0.

- The four multiplexers are controlled by two selection inputs, S1 and S0. The input carry C_{in} goes to the carry input of the FA in the least significant position. The other carries are connected from one stage to the next.

- The output of the binary adder is calculated from the following arithmetic sum:

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

where A is the 4-bit binary number at the X inputs and Y is the 4-bit binary number at the Y inputs of the binary adder. C_{in} is the input carry, which can be equal to 0 or 1

Arithmetic Circuit Function Table

S1	S0	Cin	Y	Output	Microoperation
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
0	1	1	B'	D = A + B'+ 1	Subtract With borrow
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

- When S1 S0 = 00, the value of B is applied to the Y inputs of the adder. If Cin = 0, the output D = A + B. If Cin = 1, output D = A + B + 1. Both cases perform the add micro operation with or without adding the input carry.

- When S1 S0 = 01, the complement of B is applied to the Y inputs of the adder. If Cin = 1, thenD =A +B+ 1. This produces A plus the 2's complement of B, which is equivalent to a subtract with borrow, that is, A – B – 1.

- When S1S0 = 10, the input from B are neglected, and instead, all 0's are inserted into the Y inputs. The output becomes D = A + 0 +Cin. This gives D = A when Cin = 0 and D = A +1 when Cin =1. In the first case we have a direct transfer from input A to output D. In the second case, the value of A is incremented by 1.

- When S1 S0 = 11, all 1's are inserted into the Y inputs of the adder to produce the decrement operation D = A –1 when Cin. This is because a number with all 1's is equal to the 2's complement of 1 (the 2's complement of binary 0001 is 1111). Adding a number A to the 2's complement of 1 produces F = A +2's complement of 1 = A – 1. When Cin = 1, then D = A – 1 + 1 =A, which causes a direct transfer from input A to output D.