Input-Output Organization

UNIT 4

Input-Output Interface

Asynchronous Data Transfer

Modes of Transfer

Priority Interrupt

Direct Memory Access

Input-Output Processor

Input Output Organization

- **I/O Subsystem**
  - Provides an efficient mode of communication between the central system and the outside environment

- Programs and data must be entered into computer memory for processing and results obtained from computer must be recorded and displayed to user.

- When input transferred via slow keyboard processor will be idle most of the time waiting for information to arrive
  - Magnetic tapes, disks

Peripheral Devices

- **Devices that are under direct control of computer are said to be connected on-line.**
- **Input or output devices attached to the computer are also called peripherals.**
- **There are three types of peripherals :**
  - **Input peripherals**
  - **Output peripherals**
  - **Input-output peripherals**

Peripheral (or I/O Device)

Monitor (*Visual Output Device*) : CRT, LCD,LED
KeyBoard (*Input Device*) : light pen, mouse, touch screen, joy stick, digitizer
Printer (Hard Copy Device) : **Daisy wheel**, **dot matrix** and **laser printer**
Storage Device : Magnetic tape, magnetic disk, optical disk

Q- Input or output devices attached to the computer are called

A- ITEMS
B- SORCES
C- PERIPHERALS
D- GADGETS

Input Output Organization

ASCII (*American Standard Code for Information Interchange*)

- I/O communications usually involvestransfer of alphanumeric information from the device and the computer.
- Standard binary code for alphanumeric character is **ASCII**
- ASCII Code :
  - It uses 7 bits to code 128 characters (94 printable and 34 non printing)
  - 7 bit - 00 - 7F (*0 - 127*)
- ASCII is 7 bits but most computers manipulate 8 bit quantity as a single unit called byte.

  80 - FF (128 - 255) : Greek, Italic type font
- Three types of control characters: Format effectors, Information separators and communication control

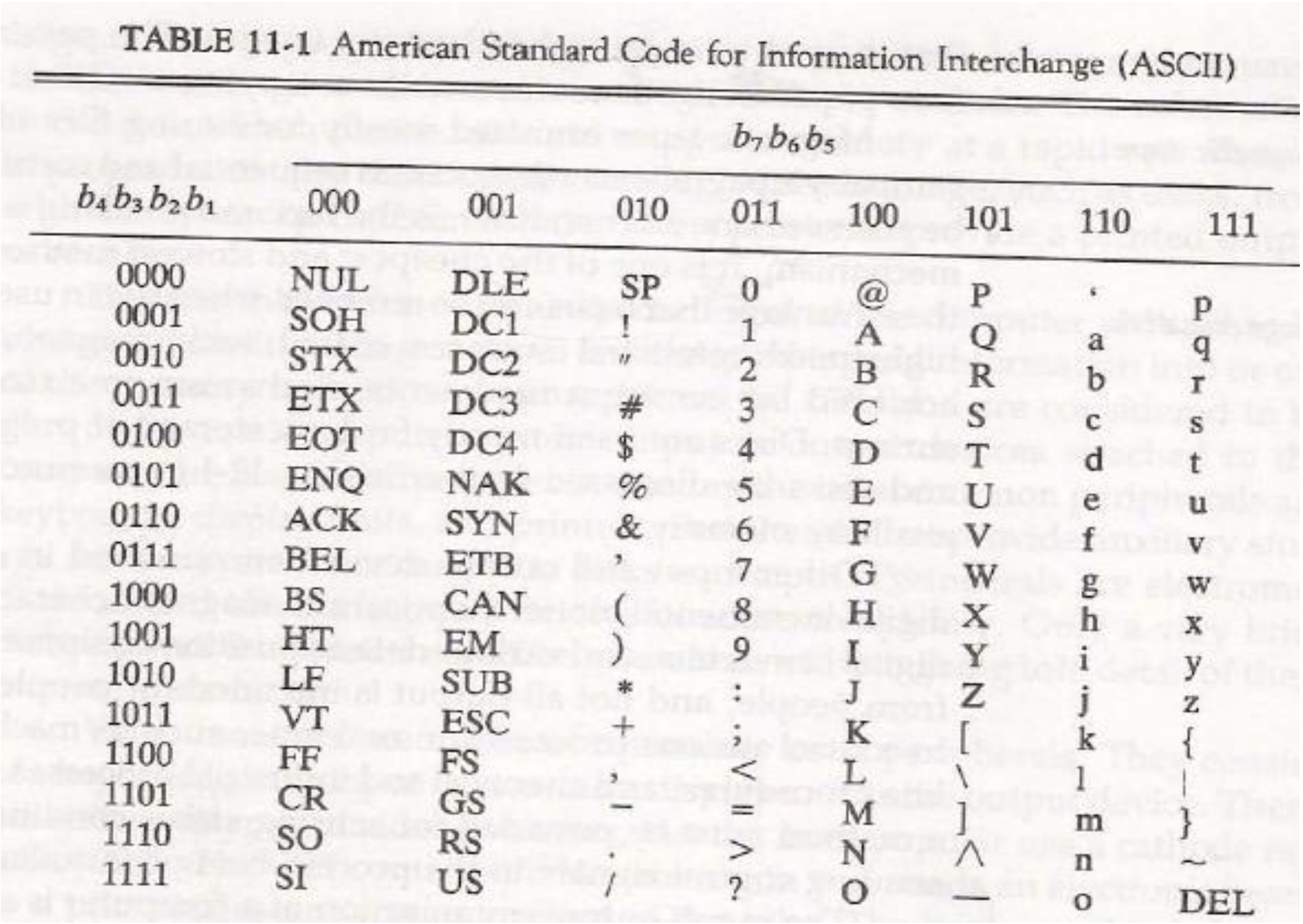
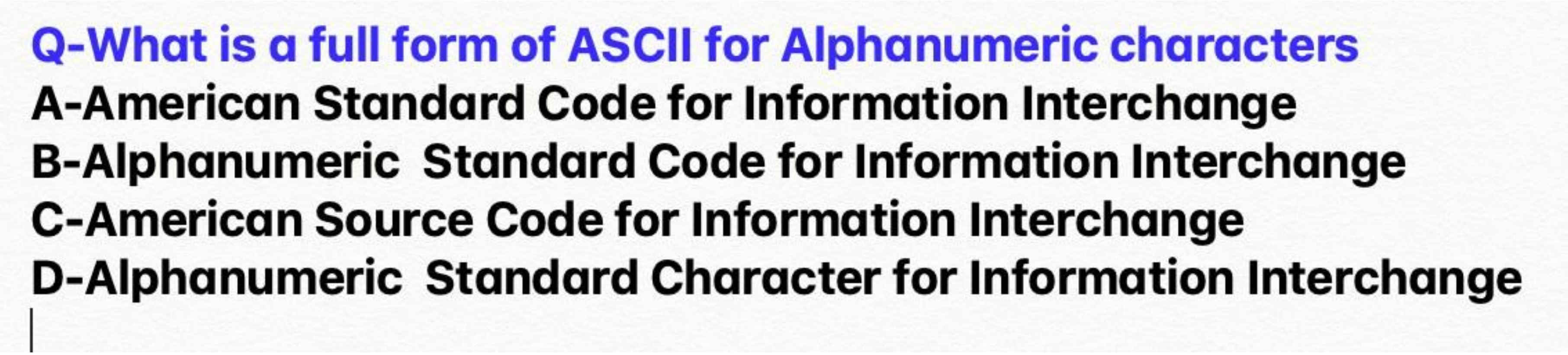


TABLE 11-1 American Standard Code for Information Interchange (ASCII)

	$b_7 b_6 b_5$							
$b_4 b_3 b_2 b_1$	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	’	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	–	=	M	]	m	}
1110	SO	RS	.	>	N	∧	n	~
1111	SI	US	/	?	O	—	o	DEL

- **Format Effectors:** control the layout of printing. They include familiar typewriter controls, such as backspace (BS), horizontal tabulation(HT), carriage return(CR).
- **Information separators:** used to separate data into divisions like paragraphs and pages. They include characters such as record separator (RS) and file separator(FS).
- **Communication Control characters**: these are useful during the transmission of text between remote terminals. These include STX(Start of text) and ETX(end of text)



Q-What is a full form of ASCII for Alphanumeric characters

A-American Standard Code for Information Interchange

B-Alphanumeric Standard Code for Information Interchange

C-American Source Code for Information Interchange

D-Alphanumeric Standard Character for Information Interchange

I/O Interface

- **Provides a method for transferring information between internal storage (such as memory and CPU registers) and external I/O devices**
- **Resolves the *differences* between the computer and peripheral devices**

  (1) Peripherals – Electromechanical or Electromagnetic Devices CPU or Memory - Electronic Device

  - **Conversion of signal values required**

  (2) Data Transfer Rate

  - **Peripherals - Usually slower**
  - **CPU or Memory - Usually faster than peripherals**
    - **Some kinds of Synchronization mechanism may be needed**

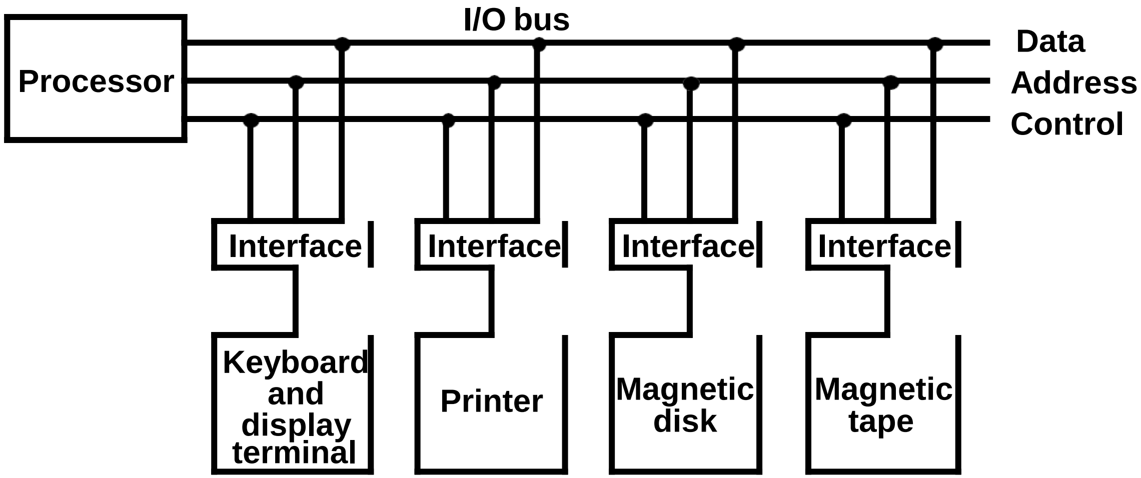
  (3) Data formats or Unit of Information

  - **Peripherals – Byte, Block, …**
  - **CPU or Memory – Word**

  (4) Operating modes of peripherals may differ

  - **must be controlled so that not to disturbed other peripherals connected to CPU**

I/O Bus and Interface



Interface :

- **Decodes the device address (device code)**
- **Decodes the commands (operation)**
- **Provides signals for the peripheral controller**
- **Synchronizes the data flow and supervises the transfer rate between peripheral and CPU or Memory**

4 types of command interface can receive : control, status, data o/p and data i/p

I/O Bus and Interface

- **Control command :** is issued to activate peripheral and to inform what to do
- **Status command :** used to test various status condition in the interface and the peripherals
- **data o/p command :** causes the interface to respond by transferringdatafrom the bus into one of its registers
- **data i/p command :** interface receives an item of data from the peripheral and places it in its buffer register.

I/O Bus and Memory Bus

Functions of Buses

- ***MEMORY BUS*** **is for information transfers between CPU and the MM**
- ***I/O BUS*** **is for information transfers between CPU and I/O devices through their I/O interface**

- **3 ways to bus can communicate with memory and I/O :**

(1) use two separate buses, one to communicate with memory and the other with I/O interfaces

- Computer has independent set of data, address and control bus one for accessing memory and another I/O.

- done in computers that have separate IOP other than CPU.

(2) Use one common bus for memory and I/O but separate control lines for each

(3) Use one common bus for memory and I/O with common control lines for both

Isolated vs. Memory Mapped I/O

Isolated I/O

-Many computers use common bus to transfer information between memory or I/O.

-The distinction between memory transfer and I/O transfer is made through separate read and write line.

-In the isolated I/O configuration , the CPU has distinct input and output instructions and each of these instruction is associated with the address of an interface register.

-Distinct input and output instructions -each associated with address of interface register

Isolated vs. Memory Mapped I/O

Memory-mapped I/O

- A single set of read/write control lines (no distinction between memory and I/O transfer)

-Memory and I/O addresses share the common address space

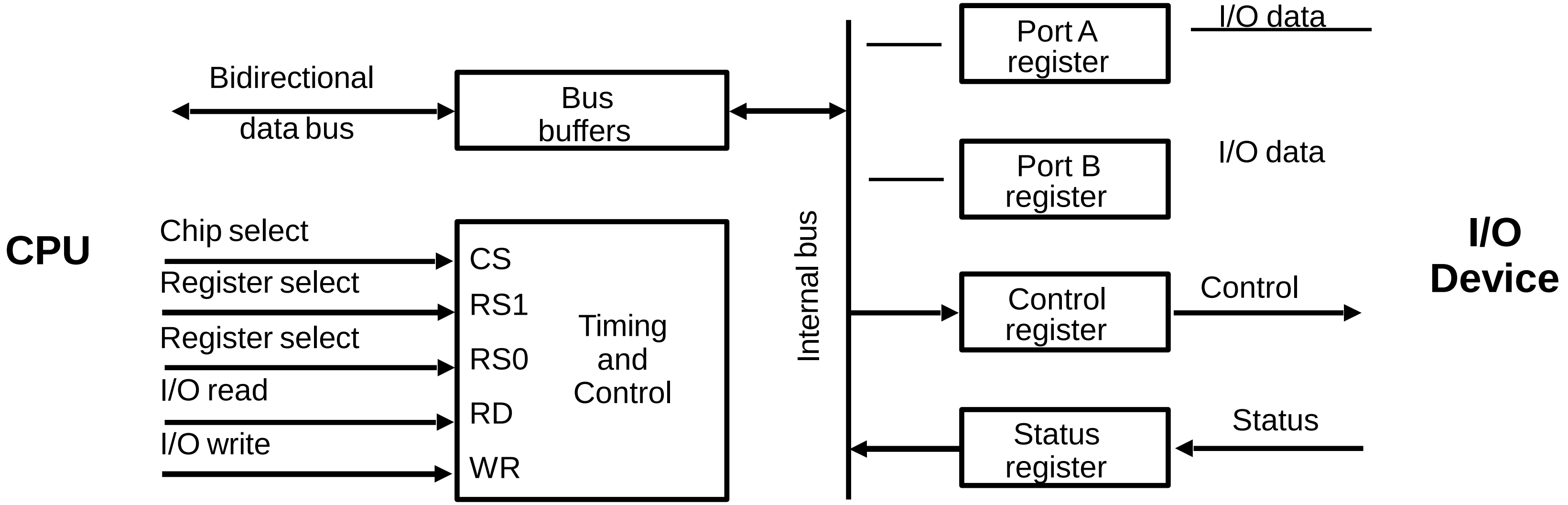
-> reduces memory address range available

- No specific input or output instruction

-> The same memory reference instructions can be used for I/O transfers

- Considerable flexibility in handling I/O operations

I/O Interface



CS	RS1	RS0	Register selected
0	x	x	None - data bus in high-impedence
1	0	0	Port A register
1	0	1	Port B register
1	1	0	Control register
1	1	1	Status register

Programmable Interface

-Information in each port can be assigned a meaning depending on the mode of operation of the I/O device→ Port A = Data; Port B = Command;

- CPU initializes(loads) each port by transferring a byte to the Control Register
→ Allows CPU can define the mode of operation of each port
→ *Programmable Port*: By changing the bits in the control register, it is possible to change the interface characteristics

The interface registers communicate with the CPU through :-

A-address bus
B-i/o bus
C-bidirectional data bus
D-control bus

There are ….. types of commands that an interface may receive.

A- 3
B-4
C-5

Data Transfer

Synchronous Data Transfer:

Clock pulses are applied to all registers within a unit and all data transfer among internal registers occur simultaneously during the occurrence of a clock pulse.

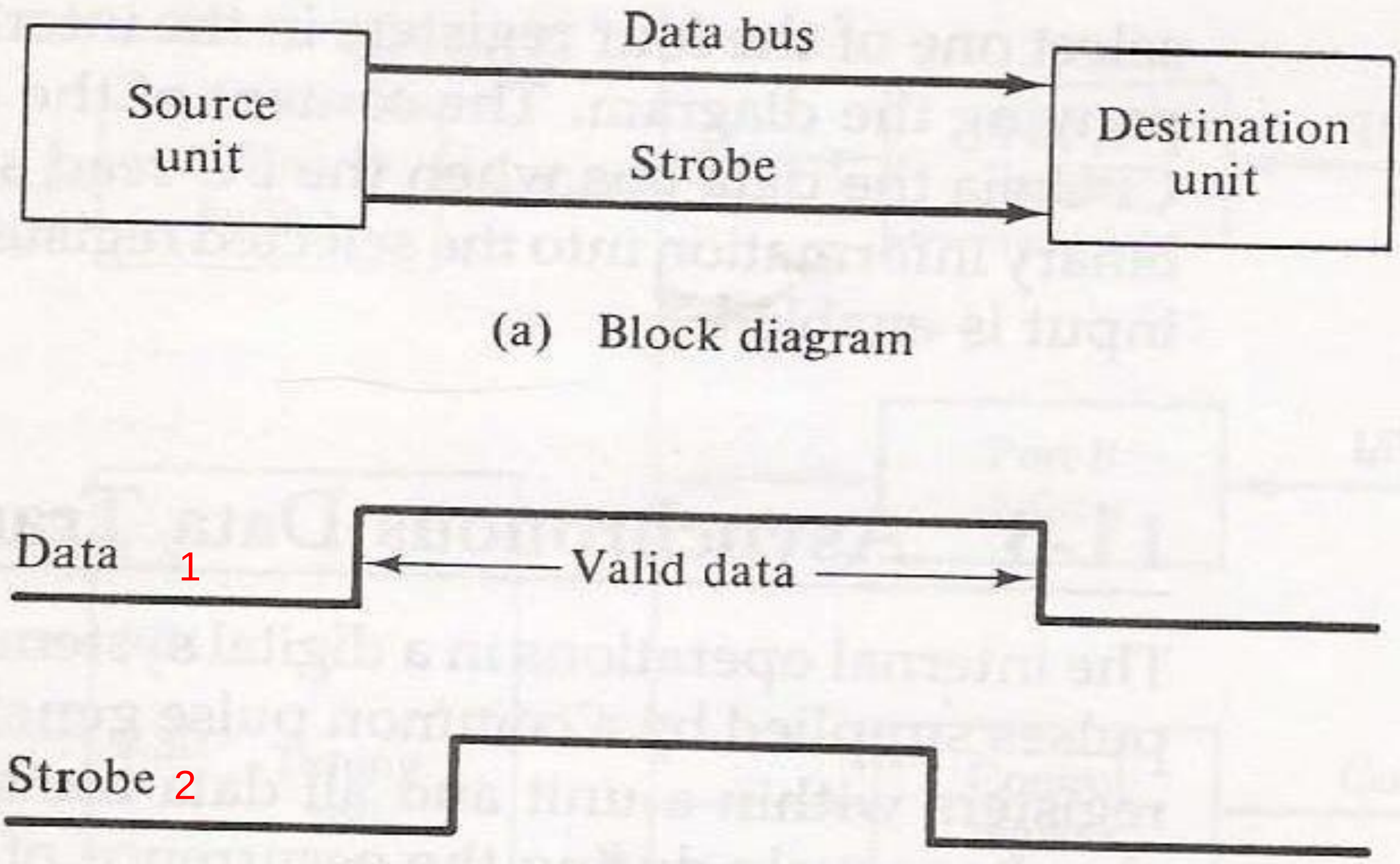
Two units such as CPU and I/O Interface are designed independently of each other. If the registers in the **interface** share a common clock with **CPU** registers, the transfer between the two is said to be **synchronous.**

Asynchronous Data Transfer:

Internal timing in each unit (*CPU and Interface*) is independent. Each unit uses its own private clock for internal registers.

Asynchronous data transfer between two independent units requires that control signals be transmitted between the communicating units to indicate the time at which data is being transmitted.

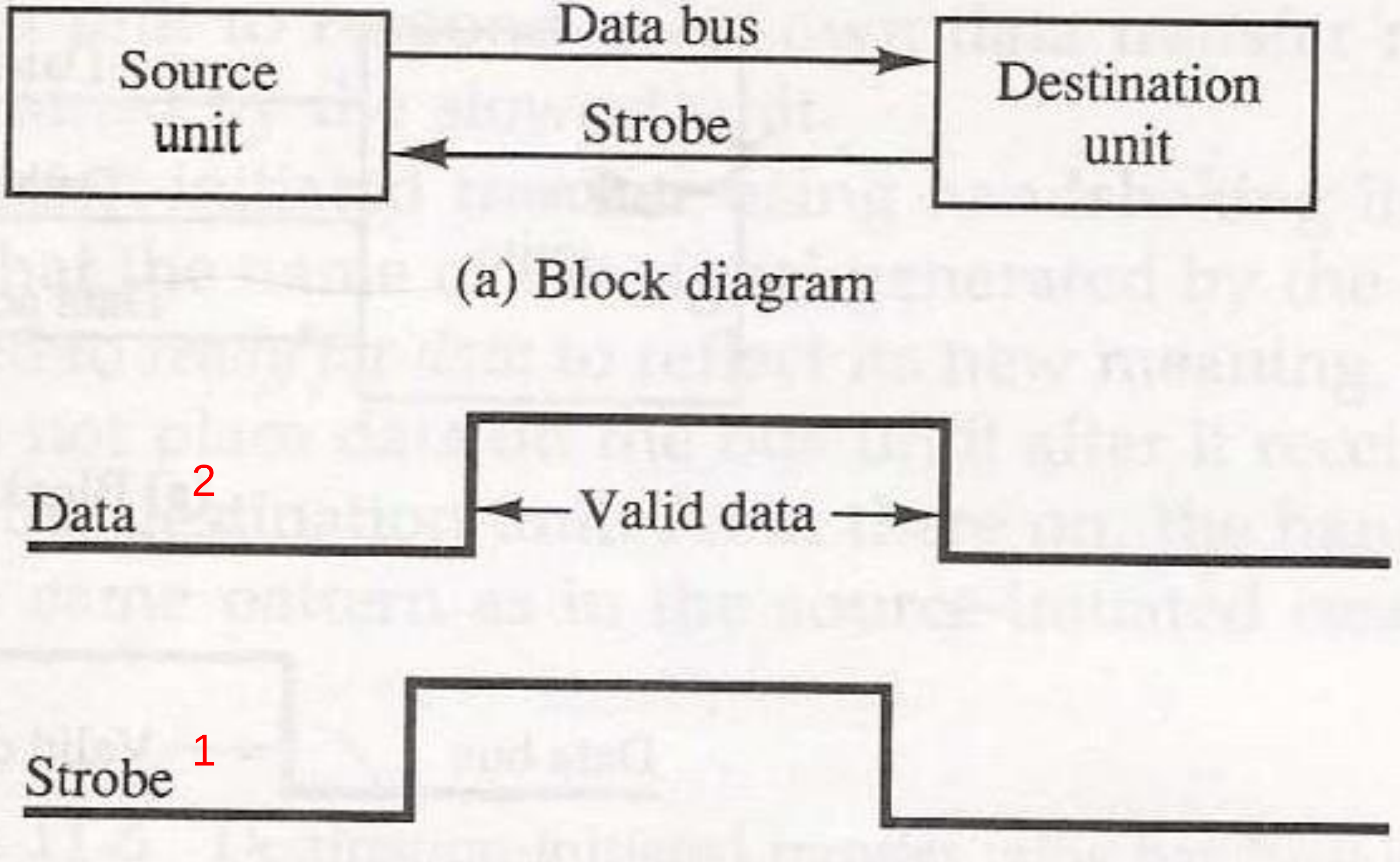
One way of achieving this is by means of **STROBE(**Control signal to indicate the time at which data is being transmitted) pulse and other method is **HANDSHAKING(**Agreement between two independent units).



(a) Block diagram

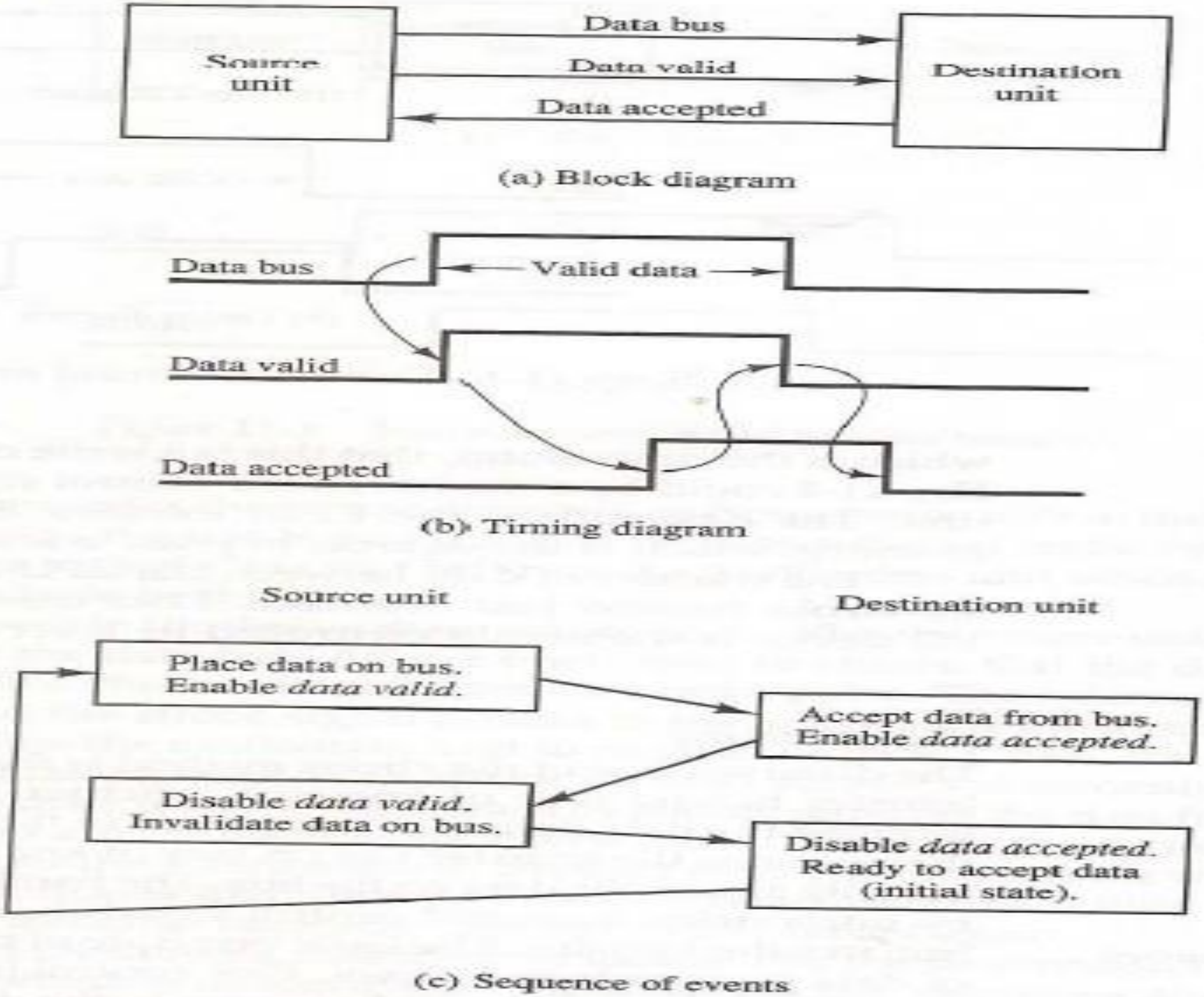
(b) Timing diagram

Figure 11-3 Source-initiated strobe for data transfer.



(b) Timing diagram

Figure 11-4 Destination-initiated strobe for data transfer.


Source unit
Data bus
Data valid
Data accepted
Destination unit
Data bus
Valid data
Data valid
Data accepted
Source unit
Destination unit
Place data on bus. Enable *data valid*.
Accept data from bus. Enable *data accepted*.
Disable *data valid*. Invalidate data on bus.
Disable *data accepted*. Ready to accept data (initial state).

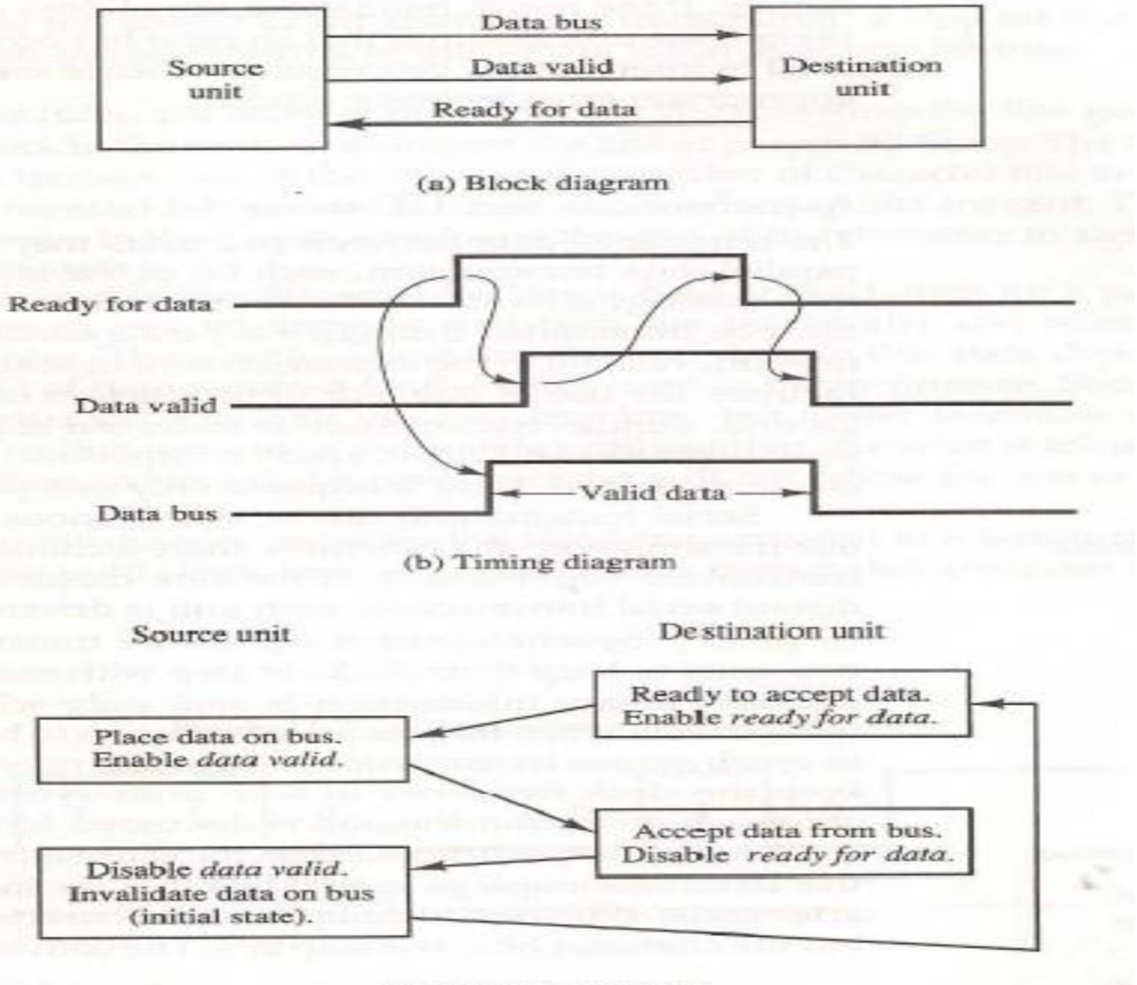
(a) Block diagram

(b) Timing diagram

(c) Sequence of events

Figure 11-5 Source-initiated transfer using handshaking.

Figure 11-6 Destination-initiated transfer using handshaking.



(a) Block diagram

(b) Timing diagram

Ready to accept data. Enable *ready for data*.

Place data on bus. Enable *data valid*.

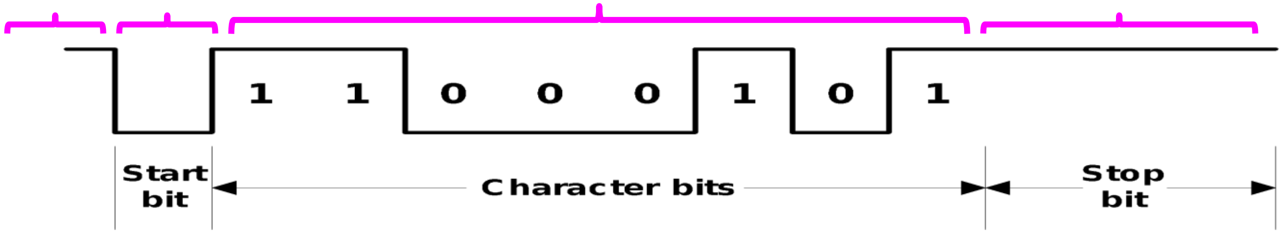
Accept data from bus. Disable *ready for data*.

Disable *data valid*. Invalidate data on bus (initial state).

(c) Sequence of events

Timeout : If the return handshake signal does not respond within a given time period, the unit assumes that an error has occurred.

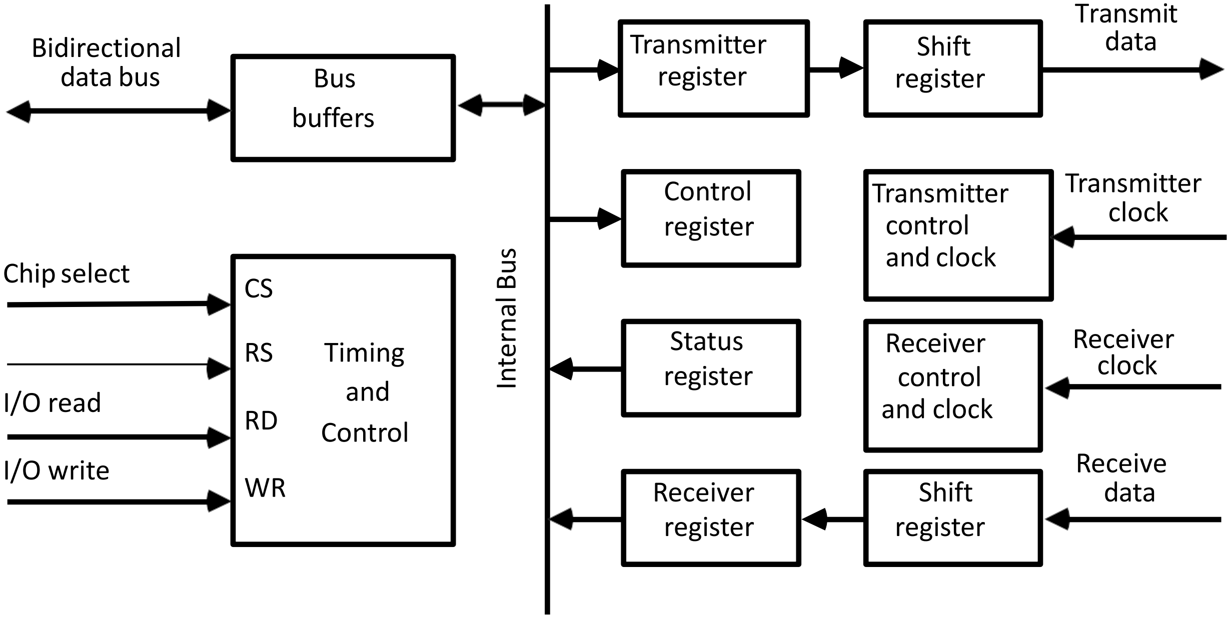
- Asynchronous Serial Transfer
  - Synchronous transmission :
    - The two unit share a common clock frequency
    - Bits are transmitted continuously at the rate dictated by the clock pulses
  - Asynchronous transmission :
    - Binary information sent only when it is available and line remain idle otherwise
    - Special bits are inserted at both ends of the character code
    - Each character consists of three parts :
      - » 1) start bit : always “0”, indicate the beginning of a character
      - » 2) character bits : data
      - » 3) stop bit : always “1”



- Asynchronous transmission rules :
  - ① When a character is not being sent, the line is kept in the 1-state
  - ② The initiation of a character transmission is detected from the start bit, which is always “0”
  - ③ The character bits always follow the start bit
  - ④ After the last bit of the character is transmitted, a stop bit is detected when the line returns to the 1-state for at least one bit time
- Baud Rate : Data transfer rate in bits per second
  - 10 character per second with 11 bit format = 110 bit per second

Universal Asynchronous Receiver Transmitter

A typical asynchronous communication interface available as an IC



CS	RS	Oper.	Register selected
0	x	x	None
1	0	WR	Transmitter register
1	1	WR	Control register
1	0	RD	Receiver register
1	1	RD	Status register

Transmitter Register

- Accepts a data byte(from CPU) through the data bus
- Transferred to a shift register for serial transmission

Receiver Register

- Receives serial information into another shift register
- Complete data byte is sent to the receiver register

Status Register Bits

- Used for I/O flags and for recording errors

Control Register Bits

- Define baud rate, no. of bits in each character, whether to generate and check parity, and no. of stop bits