CMR
INSTITUTE OF
TECHNOLOGY

USN | | | | | | | | | | |

Internal Assesment Test - II

Sub:	DSDV							Code:	15ECE663
Date:	19 / 04 / 2018	Duration:	90 mins	Max Marks:	50	Sem:	VI	Branch:	ECE

Answer the following Questions

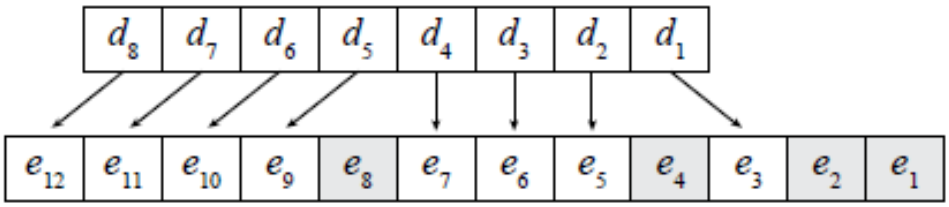
1 What is Error correction? Obtain ECC for data byte: - 01100001. Find, if received ECC 000111000100 is having error, if so correct it.

4Marks

Error-Correcting Codes (ECC)

- Allow identification of the flipped bit
- Hamming Codes
  - E.g., for single-bit-error correction of N-bit word, need $\log_2 N + 1$ extra bits
- Example: 8-bit word, $d_1 \ldots d_8$
  - 12-bit ECC code, $e_1 \ldots e_{12}$

e_1, e_2, e_4, e_8 are check bits, the rest data



$e_1 = e_3 \oplus e_5 \oplus e_7 \oplus e_9 \oplus e_{11}$

$e_2 = e_3 \oplus e_6 \oplus e_7 \oplus e_{10} \oplus e_{11}$

$e_4 = e_5 \oplus e_6 \oplus e_7 \oplus e_{12}$

$e_8 = e_9 \oplus e_{10} \oplus e_{11} \oplus e_{12}$

- Every data bit covered by two or more check bits
- On write: Compute check bits and store with data
- On read: Recompute check bits and XOR with read check bits
  - result called the *syndrome*
- 0000 => no error
- If data bit flipped
  - covering bits of syndrome are 1
  - = binary code of flipped ECC bit
- If stored check bit flipped
  - that bit of syndrome is 1
- On error, unflip bit and rewrite memory location

ii) The check bits are

$e_1 = e_3 \oplus e_5 \oplus e_7 \oplus e_9 \oplus e_{11} = d_1 \oplus d_2 \oplus d_4 \oplus d_5 \oplus d_7 = 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 = 0$

$e_2 = e_3 \oplus e_6 \oplus e_7 \oplus e_{10} \oplus e_{11} = d_1 \oplus d_3 \oplus d_4 \oplus d_6 \oplus d_7 = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 = 1$

$e_4 = e_5 \oplus e_6 \oplus e_7 \oplus e_{12} = d_2 \oplus d_3 \oplus d_4 \oplus d_8 = 0 \oplus 0 \oplus 0 \oplus 0 = 0$

$e_8 = e_9 \oplus e_{10} \oplus e_{11} \oplus e_{12} = d_5 \oplus d_6 \oplus d_7 \oplus d_8 = 0 \oplus 1 \oplus 1 \oplus 0 = 0$

Thus the ECC word is 011000000110.

3Marks

iii) Determine whether there is an error in the ECC word 000111000100, and if so, correct it.
solution The check bits computed from the data bits of the ECC word are

$e_1 = e_3 \oplus e_5 \oplus e_7 \oplus e_9 \oplus e_{11} = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1$

$e_2 = e_3 \oplus e_6 \oplus e_7 \oplus e_{10} \oplus e_{11} = 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 0$

$e_4 = e_5 \oplus e_6 \oplus e_7 \oplus e_{12} = 0 \oplus 0 \oplus 1 \oplus 0 = 1$

$e_8 = e_9 \oplus e_{10} \oplus e_{11} \oplus e_{12} = 0 \oplus 0 \oplus 0 \oplus 1 = 1$

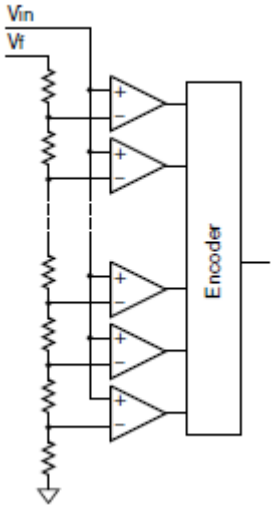
3Marks

The syndrome is 1101 ⊕1000 _ 0101. Thus, there is an error in bit e5 of the read ECC. That bit should be flipped back from 0 to 1, giving the corrected ECC word 000111010100.

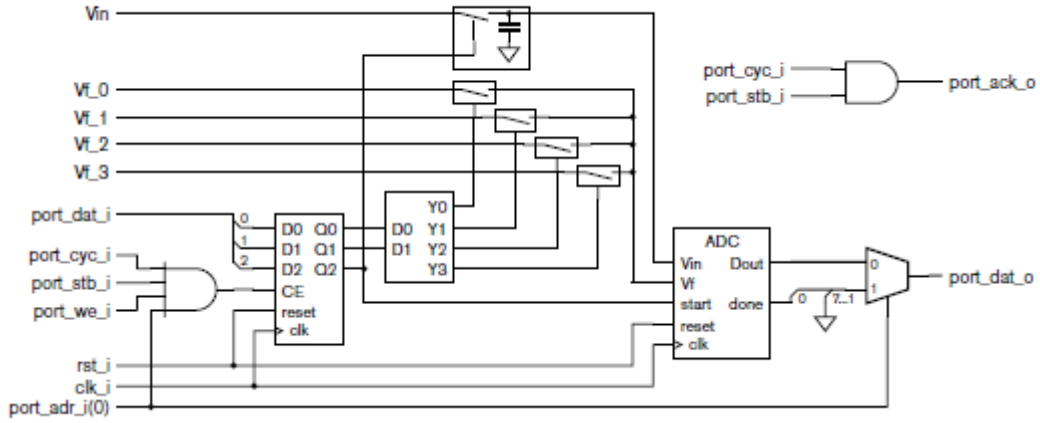
2 Briefly explain the Serial Interface Standards for I/O device.

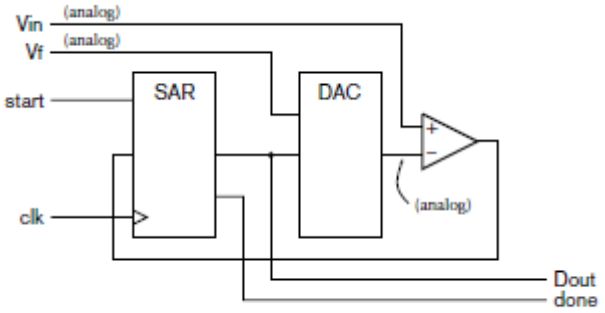
RS-232: This standard was originally defined in the 1960s for connecting teletype computer terminals with modems, devices for serial communication with remote computers via phone lines. Sub sequently, the standard was adopted for direct connection of terminals to computers. Since most computers included RS232 connection

	ports, RS232 connections were incorporated in I/O devices other than terminals as a convenient way to connect to computers. Examples included user-interface devices such as mice, and various measurement devices. Serial transmission in RS232 interfaces uses NRZ encoding with start and stop bits for synchronization. Data is usually transmitted with the least significant bit first and most significant bit last. While RS232 interfaces have now largely been supplanted by more recent standards, they are still used in some equipment, for example, bar code readers in point-of-sale terminals, and industrial measurement devices. I2C: The Inter-Integrated Circuit bus specification is defined by Philips Semiconductors, and is widely adopted. It specifies a serial bus protocol for low-bandwidth transmission between chips in a system (10kbit/sec to 3.4Mbit/sec, depending on the mode of operation). It requires two signals, one for NRZ-coded serial data and the other for a clock. The signals are driven by open-drain drivers, allowing any of the chips connected to the bus to take charge by driving the clock and data signals. The specification defines particular sequences of logic levels to be driven on the signals to arbitrate to see which device takes charge and to perform various bus operations. The advantage of the I2C bus is its simplicity and low implementation cost in applications that do not have high performance requirements. It is used in many off-the-shelf consumer and industrial control chips as the means for an embedded microcontroller to control operation of the chip. Philips Semiconductor has also developed a related bus specification, I2S, or Inter-IC Sound, for serial transmission of digitally encoded audio signals between chips, for example, within a CD player. USB: The Universal Serial Bus is specified by the USB Implementers Forum, Inc., a nonprofit consortium of companies founded by the original developers of the bus specification. USB has become commonplace for connecting I/O devices to computers. It uses differential signaling (see Section 6.4.1) on a pair of wires, with a modified form of NRZ encoding. Different configurations support serial transfer at 1.5Mbit/sec, 12Mbit/sec or 480Mbit/sec. The USB specification defines a rich set of features for devices to communicate with host controllers. Since there is such a diversity of devices with USB interfaces, application-specific digital systems can benefit from inclusion of a USB host controller to enable connection of off-theshelf devices. USB interface designs for inclusion in ASIC and FPGA designs are available in component libraries from vendors. FireWire: This is another high-speed bus defined by IEEE Standard 1394. Whereas USB was originally developed for lower bandwidth devices and subsequently revised to provide higher bandwidth, FireWire started out as a high-speed (400Mbit/sec) bus. There is also a revision of the standard defining transfer at rates up to 3.2Gbit/sec. FireWire connections use two differential signaling pairs, one for data and the other for synchronization. As with USB, there is a rich set of bus operations that can be performed to transmit information among devices on the bus. FireWire assumes that any device connected to the bus can take charge of operation, whereas USB requires a single host controller. Thus, there are some differences in the operations provided by FireWire and USB, and some differences in the applications for which they are suitable. FireWire has been most successful in applications requiring high-speed transfer of bulk data, for example, digital video streams from cameras.
3	Explain the working of a)Flash ADC b) Successive approximation ADC



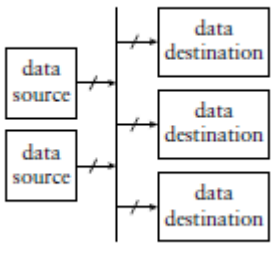
Flash ADC





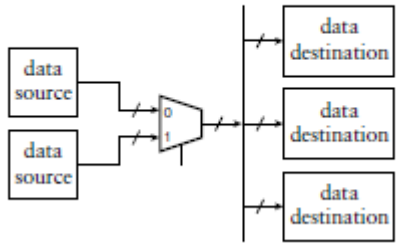
SAR based ADC

4	Explain the parallel bus concept. Briefly discuss the solutions to overcome the disadvantages of parallel buses.

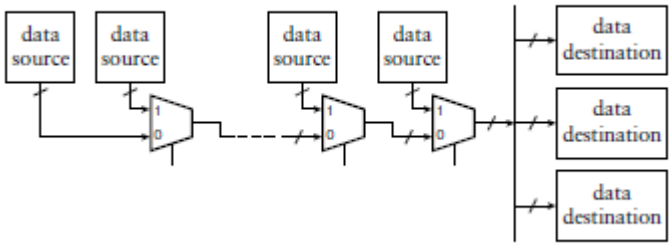


Multiplexed buses

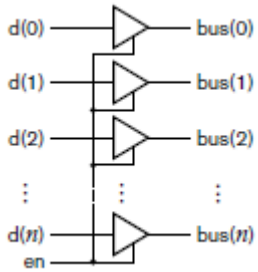
8.3 Parallel



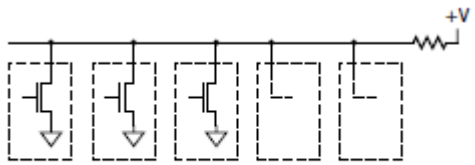
Distributed multiplexed bus



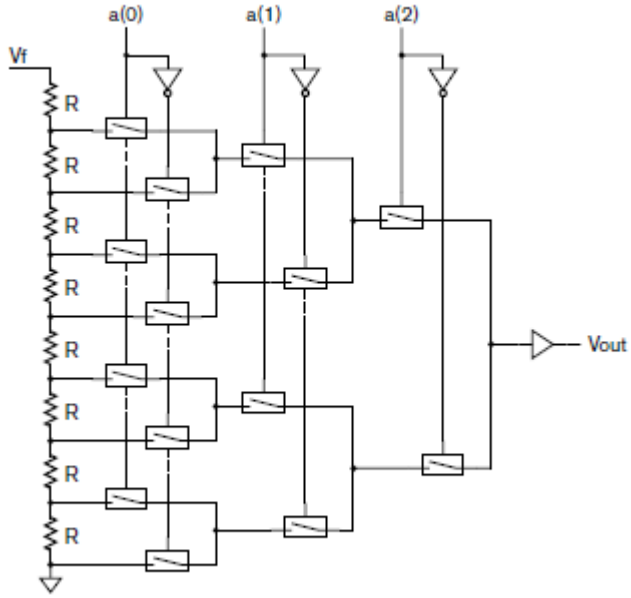
Tristate buses



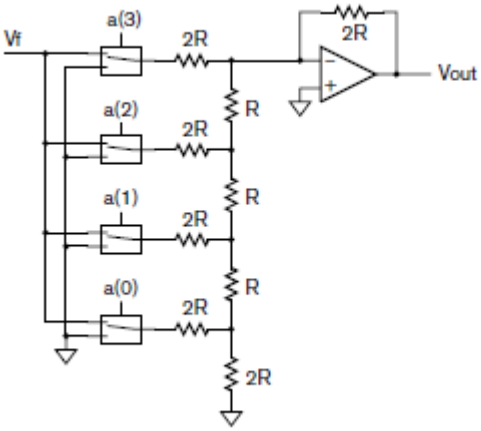
Open drain buses



5	What is a DAC? Explain different forms of DAC.



R string DAC



R-2R DAC