CMR
INSTITUTE OF
TECHNOLOGY

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

Internal Assesment Test – III

Sub:	DSDV							Code:	15EC663
Date:	15/ 05 / 2019	Duration:	90 mins	Max Marks:	50	Sem:	VI	Branch:	Open Elective

Answer all Questions

		Marks	OBE CO	OBE RBT
1	What are the serial interface standards? Explain any four. **Solution:-** These standards cover two broad areas of serial interfaces: connection of I/O devices to computers, and connection of computers together to form a network. Some examples of serial interface standards for connecting I/O devices include: i) 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 * Subsequently, the standard was adopted for direct connection of terminals to computers. Since most computers included RS232 connection ports, RS 232 connections were incorporated in I/O devices other than terminals as a convinient way to connect to computers. * Serial transmission in RS232 interfaces uses NRZ encoding with start and stop bits for synchronization with LSB first and MSB at last. ii) I²C * The Inter Integrated circuit bus specification is defined by Philips Semiconductors. and is widely adopted * It specifies a serial bus protocal for low bandwidth transmiss between chips in a system * It requires two signals, one for NRZ coded serial data and the other for a clock	[10]	CO4	L2

	* 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 advantage of I²C bus is its simplicity and low implementation cost in applications that do not have high performance requirements iii) 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 on a pair of wires with a modified form of NRZ encoding. * Different configurations support serial transfer at 1.5 Mbit/sec, 12 Mbit/sec or 480 Mbit/sec iv) Firewire: * This is another high speed bus defined by IEEE Standards 1394. whereas USB was originally developed for lower bandwidth devices and subsequently revised to provide higher bandwidth, Firewire started out as high speed bus. * There is also a version of the standard defining transfer at rates up to 3.2 Gbit/sec * Firewire connections use two differential signaling pairs one for data and other for synchronization * Firewire has been most successful in applications requiring high-speed transfer of bulk data for example, digital video streams from cameras.			
2	Write a note on polling and timers **Solution:-**	[10]	CO4	L3

POLLING:

* The simple I/O sychronization is called polling
* It involves the software repeatedly checking a status input from a controller to see if an event as occured.
* If it as, the software performs the necessary task to react to the event
* If there are multiple controllers, or multiple events to which the software must respond, the software checks each of the status inputs in tiun, reacting to eve as they occur, as part of a busy loop
* Polling has the advantage that is very simple to implement and requires no additional circuitary beyond the input and output registeas of the I/O controllers.

	* However, it requires that the processor core be continually active, consuming power even when there is no event to react to. * It also prevents the processor from reacting immediately to one event if it is busy dealing with another event. For these reasons, polling is usually only used in very simple control applications where there is no need for fast reaction times. TIMERS: * Many real time embedded systems must perform actions at specific times. To these systems we need to include some form of timer. * We can use counter to derive a periodic signal from system clock. We can use such a signal as a time base: each cycle represents one unit of time in embedded system. * A common use for interval timers in real time embedded systems is to generate an interrupt for the processor at some programmable multiple of a time base. * The interval timer acts as an I/O controller often called a real-time clock, with an output register for programming the time interval. * The interrupt handler for the timer can then perform any required periodic actions * An RTOS generally includes an executive, together with software components to manage other resources such as storage, input/output communication and specialized processing resources. * The advantage of using RTOS is that we can focus our software development effort on the aspects of our system that are different from other systems and reuse proven code that deals with common embedded software mechanisms.			
3	Explain Functional verification, synthesis and design optimization. **Solution:-**	[10]	CO5	L2

optimization

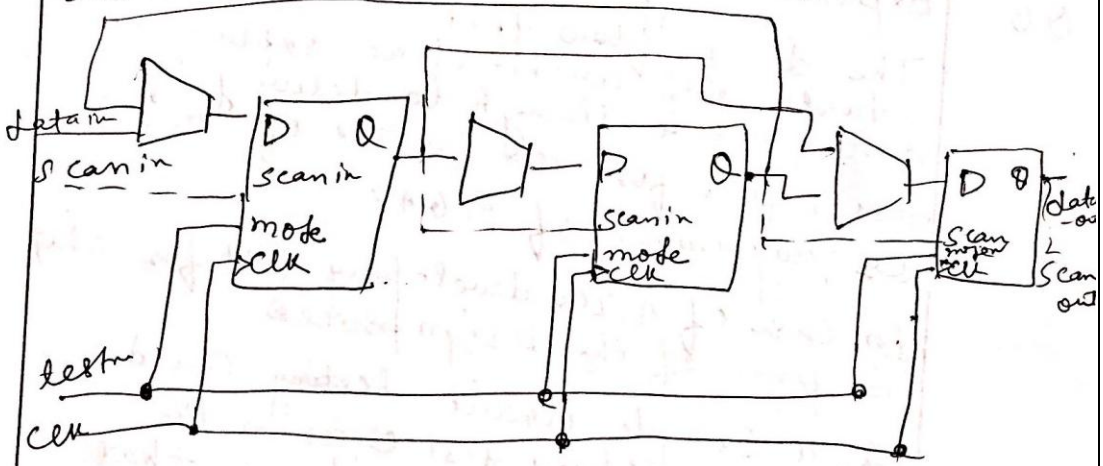
Functional Verification :- Successful Verification of a system requires a verification plan that identifies what part of the design will be verified, the functionality that will be verified and how verification will be performed. Since system is composed of subsystems, each subsystems must be correct for the entire system is composed of subsystems, each subsystems must be correct for entire system to be correct.

We use term coverage to refer the proportion of functionality that is verified. Code coverage has been used a figure of merit. It refers to proportion of lines of code that have been executed to at least once during simulation of design.

SYNTHESIS - having performed functional design and verification of digital system, the next stage in the design flow is synthesis, that is refinement of the functional design to a gate level net list. For most design, Synthesis can be performed largely automatically using an RTL Synthesis tool. RTL Synthesis, as the name suggests, start with models of design refined to register transfer level.

	Design optimization - We described a design flow assuming that the design meets constraints at each stage. In most design projects, this ideal situation does not hold. We need to perform some optimization of the design, possible making trade-off of one property against another. Instead they are cyclic with the design evolving as more "back-end" implementation. & if fine-tuning is insufficient, then we need to revisit earlier stages to make more substantial changes.			
4	Explain scan design and boundary scan **Solution:-**	[10]	CO5	L3

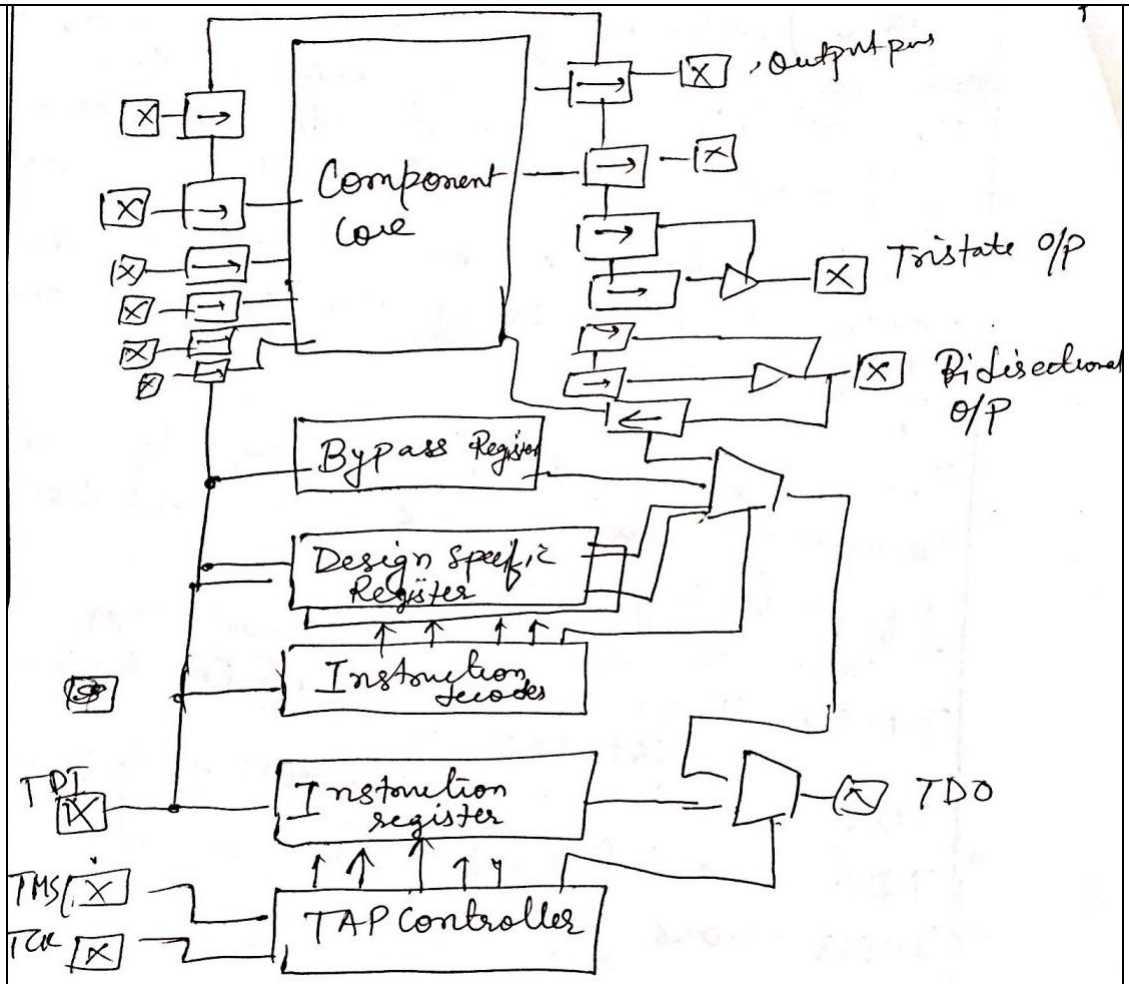
Scan design:- Scan design technique address modifying the registers to allow them to be chained into a long shift registers called scan chain. Test vectors can be shifted into the registers in the chain, under control of test mode input, making them controllable. Stored values can also be shifted out the registers, thus making them observable. We then run the system normal operation mode for one clock cycle, clocking the output of each block into ~~block~~ blocks output registers



boundary Scan - The concept of scan design can be extended for using in testing the connection between the chips on PCB. To test the PCB to include scan-chain flip flops on the external pins of each chip. When the chain is loaded the vector is driven on to the external outputs of the chips

The JTag Standard Specifies the each component have Access port, considering of following Connection

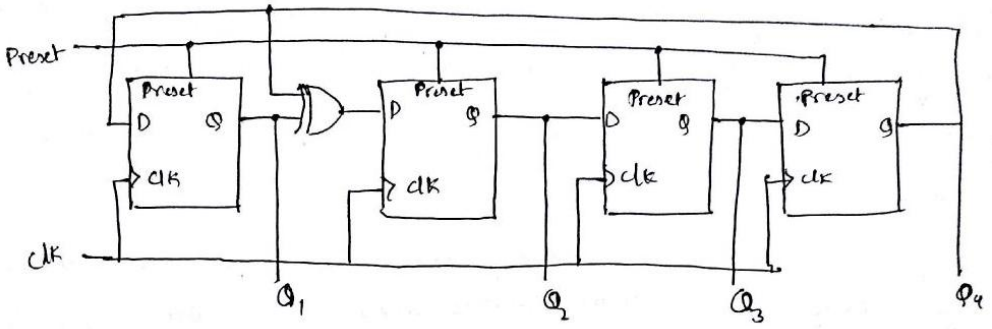
TCK → control test operation (TCK)
TMS → Test modes select Input control test operation
TDI → Serial input for test data & instruction
TDO → Serial output for test data and instructions:

				
5	Define DFT and BIST, Design a 4 bit LFSR for generating pseudo-random test vectors. **Solution:-**	[10]	CO5	L3

a) DFT: Design for test which refer to design techniques that seeks to improve testability.

BIST: Built-in self test, techniques which involve adding test circuits that generates test patterns and analyze output responses.

A 4-bit LFSR for generating pseudo-random test vectors



1111 → 1011 → 1001 → 1000 → 0100 → 0010 → 0001 → 1100

0111 ← 1110 ← 0101 ← 1010 ← 1101 ← 0011 ← 0110

The most common means of generating test patterns is a pseudo random test pattern generator. Unlike true random sequences, pseudo-random sequences can be repeated from a given starting point, called the seed.

They have similar statistical properties to true random sequences. Pseudo-random sequences can be readily generated with a simple hardware structure called linear-feedback shift register.

The sequence is initiated by presetting the flip flops generating the test value 1111 as a seed.