viva

Embedded
↳ sth hidden

Chap. 1 Introduction.

① ES defⁿ? Why is it hard to define?

⇒ An embedded system is nearly any computing system other than a desktop computer.
ES hard to define because they cover such a board range of electronic devices.

② Characteristics of E.S.

i) single functioned: executes a specific program repeatedly
ii) tightly constrained: the specified cost, performance, and power must be met (meet શર્તો)
iii) Reactive & real time system:-
system ને outside stimuli ને quickly & constantly react કરવું પડે.

③ Classification of ES

Small scale ES	Medium scale ES	Sophisticated
↓	↓	↓
single 8 bit / 16-bit microcontroller	single or few 16/32 bit microcontroller. DSP, RISC.	Enormous H/w & S/w complexity
eg. washing system, printer, scanner	eg. Banking systems like ATM	eg. Real time video processing system, Speech or multimedia processing system.

Embedded System viva question collection

Khom

(4) Essential components of ES.
- Embeds HW → processor, Timers, ...
- Embeds main app^n s/w.
- Embeds a real time O.S.

(5) Overview of processors in ES.
(A) GPP (General purpose processor)
i) microprocessor : Intel : 8085, 8086, ...
Motorola : 6800, 6809, ...
ii) Microcontroller : Intel : 8032, 8051, 8052
PIC : 8-bit PIC, ...
Motorola : 6811
iii) Embedded processor
iv) Digital signal processor

(B) Application Specific processor
- dedicated to specific tasks like image compression & provides faster sol^n.
e.g. IIM7100, W3100A

(C) Single-purpose processor
- Digital ckt designed to execute only a single program.
eg. coprocessor, accelerator or peripheral.

(6) Hardware units in E.S

(7) Application of E.S
- (i) Automotive Electronics
- (ii) Aircraft electronics
- (iii) Trains
- (iv) medical systems
- (v) Robotics
- (vi) military applications.

(8) Design metrics.
- cost → chip cost [illegible],
- power → power consumed by system, low [illegible]
- size → physical space in bytes or gates required
- performance → Execution time of system.

Embedded System viva question collection

Khom

Chap 2 Hardware & Software Design Issues :-

(1) What is a single-purpose processor? What are the benefits of choosing a single-purpose processor over a general-purpose processor.

=) A single purpose processor is designed specifically to carry out a particular computational task

Benefits of choosing single-purpose processor over general-purpose processor is that
- the performance will be faster,
- size smaller,
- power consumption less

(2) How do nMOS & pMOS transistors differ?

nMOS conducts if gate is high (i.e. 1) } complement
&
pMOS conducts if gate = 0 (low) }

(3) NAND & NOR gates more common than AND and OR gates due to their reduced size.

(4) Combinational logic ckt =)
→ is a type of logic ckt whose o/p is a function of only the present input

(5) Sequential logic ckt
→ is a type of logic ckt whose o/p depends not only on present values of its i/p signals but on the sequence of past i/p's, as well.

CMOS - complementary metal oxide semiconductor

(6) Transistor :- basic electrical component in digital systems.
- Acts as on/off switch.

(7) Basic gates :-
inverter, AND, NOR.

(8) Differentiate combinational & sequential logic design

Combinational	Sequential
① O/p determined by present values of i/p only.	① It's o/p is determined by the present values of i/p as well as past values of o/p.
② Combinational does not have memory	② Seq. has memory
③ no feedback path from o/p to i/p.	③ It has feedback path
④ eg. Adder, subtractor, mux, demux, decoder,	④ eg. flipflops, counters, shift registers.

(9) GCD (Greatest common Divisor) of two no.
→ is the largest no. that divides both of them

(10) FSMD → finite state machine with datapath.

(11) GCD :-
→ first create algorithm.
→ convert algorithm to "complex" state machine
- FSMD
- templates use કરી ને conversion.

(12) Datapath માટે create કરો
→ create register ને declared variable
→ create a functional unit ને each arithmetic op'n
→ connect ports, registers & functional units.

→ create unique identifier.
for each i/p & o/p.

(13) Creating Controller's FSM :-
- same as FSMD
- Replace complex actions/conditions with datapath configurations.

(14) Function of controller (purpose)
→ moves data through datapath.
→ controlling of data

(15) Datapath → stores & manipulates data.
↳ consists of
i) Register units like Reg A, B, - -
ii) functional "
iii) connection "

(16) Optimization :- is the task of making design metrics values the best possible.

(17) Optimization opportunities
- original program
- FSMD
- datapath
- FSM

(18) Optimizing the original program :-
→ Analyze program attributes and look for areas of possible improvement.
- no. of computations
- size of variable,
- time & space complexity
- operations used
  → multiplication & division very expensive.

(19)

S/W Design Issues : GPP.

(1) The general purpose basic architecture consists of :
- Datapath
- Control unit
- memory

(2) Datapath → consists the circuitry to transfer the datapath from one place to another & storing temporary data

(3) Control unit

Harvard	Princeton
→ separate memory space to store program as well as data & requires different connection. eg. microcontroller 80.51/52	→ share common memory space to store data & program and requires one to one connection with n/w

(4) Programmer's view :-

A programmer writes the program instruction to carryout the desired functionality on GPP.

3 levels of programming

(1) Assembly level → low-level, assembly code
(2) Structure " → logical programming, OOP.
(3) Machine " → machine code or object code.

(5) OS

→ interface betⁿ a computer user and computer h/w.

Software → System s/w + Application s/w

O.S.

Hardware → CPU, RAM, I/O

(6) API (Application programming Interface)

Benefits

portability : supportive .- - compile & run

Ease of use :

(7) Devᵐⁿᵗ Envᵐⁿᵗ

comprise of general s/w tools they are used to design, testing, validation & verification of ES s/w.

(8) IDE :

that assists programmers in developing s/w.

eg. GNU Compiler collection (GCC), Eclipse, Dephi, PyCharm.

(9) Editor :

A source code editor is a text editor program designed specifically for editing source code to control embedded systems.

(10) Compiler :

→ It is a computer program that translates the source code into computer language (object code).

(11) Linker :- It is a program that takes one or more objects generated by compilers & assembles them into a single executable program or a library that can later be linked to in itself.

(12) Debugger :- It is a computer program that is used to test & debug other programs.

Chap 3

Memory :-

(1) → Stores instr^ns as well as data
The memory is connected to the processor through the following lines.
i) Address
ii) Data
iii) Control

(2) Memory at basic level classification.
i) Processor memory (Register Array)
ii) Internal on-chip memory
iii) Primary memory
iv) Secondary memory

(3) Memory write Ability -
process of putting the bits in specific location of the memory & the ease and speed with which the process can be completed.

(4) Storage permanence
- ability to hold the stored bits.
- The ability is temporary or permanent.

khom

(5) Common Types of Memory

A) Mask-programmed ROM:
programming done at fabrication time with masks.

B) PROM:
User programmable ROM is memory that can be written after fabrication but before normal opr^{n}

C) EPROM:
Erasable PROM can be programmed by injecting e⊖s into "floating gates".

D) EEPROM:
Electrically-erasable programmable ROM's are programmed & erased electronically,

E) Flash EEPROM
Flash EEPROM's - able to erase large blocks of memory rather than just a word at a time although they can be slower at writing single words

(6)

(6) RAM :-

i) SRAM :-

Static RAM holds data for as long as there is power supplied to it.

It is typically implemented on the IC using Flip-flops to store bits

ii) DRAM :-

smaller than SRAM because it uses only a MOS transistor and a capacitor to store a bit.

(7) Composing memory ->

(8) Memory Hierarchy & Cache :

Chap. (4) Interfacing

(1) Difference betn.

Memory mapped I/O	I/O Mapped I/O
(i) uses same address space for both memory & IO device.	(i) uses two separate address spaces for memory & IO device.
(ii) only 1 set of read & write instruction lines.	(ii) I/O read & I/O write lines for I/O transfer.
(iii) No separate instruction like IN, OUT & MOV.	(iii) IN & OUT instructions deals with I/O transfer.

(2) DMA

→ Direct memory Access

is a process of transferring data from one memory location to another without the direct involvement of the processor (CPU).

main benefit

→ more efficient data movement in E.S.

(3) Arbitration :-

Bus arbitration refers to process by which the current bus master accesses & then leaves the control of the bus & passes it to another bus requesting processer unit.

④ DMA controller.

→ allows device to transfer the data directly to/from memory without any interference of the CPU.

Using DMA controller, the device requests the CPU to hold its data, address & control bus, so the device is free to transfer data directly to/from the memory.

⑤ Arbitration: Priority Arbiter (Daisy chain)

Priority Arbiter.

→ Single-purpose processor.

→ peripherals make requests to arbiter, arbiter makes requests for resource.

→ Arbiter connected to the system bus for configuration only

⑥ Multilevel Bus Architectures:-

दोनों one bus for all communication use होने से होने

→ peripherals would need high-speed, processor-specific bus interface.

⑦ processor-local bus.

→ high speed, wide, most frequent

→ communication.

→ connects microprocessor, cache, memory, controllers, etc.

(8) Peripheral bus.
→ lower speed, narrower, less frequent comm^n.
→ typically industry standard bus for portability

(9) Bridge ?
→ single-purpose processor converts communication between buses

(10) Advanced Communication principles.
Layering थरोंका

Break complexity of communication protocol into pieces easier to design & understand.

(11) Parallel communication :
multiple bits of data एक-साथ भेजा जाता है

(12) serial communication :
one bit of data at a time transfer होता है

(13) wireless communication
No physical connection needed for transport a physical layer.

parallel 2 serial में physical layer same है

(14) Error detection & correction.

Error detection : ability of receiver to detect errors during transmission.

Error correction : ability of receiver and transmitter to cooperate to correct problem
↳ typically done by acknowledgement/retransmission protocol.

(15) Bit error : single bit invert error

(16) Burst of bit error : consecutive bits received incorrectly

(17) Parity : extra bit that is sent with word used for error detection.

(18) Parity types
a) Odd parity : data word plus parity bit contains odd no. of 1's
b) Even parity : data word plus parity bit contains even no. of 1's

(c) Always detects single bit errors, but not all burst bit errors.

(19) Checksum :- extra word sent with data packet of multiple words.
eg. extra word contains XOR sum of all data words in packet

(20) I^2C (Inter-IC)
→ Two-wire serial bus protocol developed by philips Semiconductors nearly 20 years ago.
→ enables peripheral ICs to communicate using simple comm^n h/w.
→ Data transfer rates up to 100 kbits/s & 7-bit addressing possible in normal mode.

(21) I^2C Bus Structure दोनों तरह

→ micro controller (master) एक

→ EEPROM (serial)

→ Temp. Sensor (serial)

→ LCD-controller (serial)

(22) Serial protocols :

i) CAN (controller Area network)
→ protocol for real-time applications.
→ developed by Robert Bosch GmbH

ii) Fire Wire (also known as I-link, Lynx, IEEE 1394)
→ High-performance serial bus developed by Apple Computer Inc.
→ Designed for interfacing independent electronic components eg. Desktop, scanner

iii) USB (Universal Serial Bus)
→ Easier connection betn PC & monitors, printers, digital speakers, modems, ---

(23) Parallel protocols :

i) PCI Bus. (Peripheral component Interconnected)
→ High performance bus originated at Intel in the early 1990's

ii) ARM Bus
→ Designed & used internally by ARM corporation.
→ 32-bit addressing

iii) IrDA (Infrared Data Association)
→ protocol suite that supports short-range point-to-point infrared data transmission

(24) Wireless protocols:

i) Bluetooth:

- Daily life all use wireless.
- New, global standard for wireless connectivity
- low cost

ii) IEEE 802.11

proposed standard for wireless LANs

Chap 5 · Real Time Operating System (RTOS)

① Desktop OS Vs RTOS

Desktop OS	RTOS
i) General purpose	i) Dedicated to a single embedded application.
ii) During boot up, OS gets control first then application s/w.	ii) During boot up, Application s/w gets the control first then the OS.
iii) Application & OS share the different address space	iii) Application & OS share the same address space.
iv) eg. windows, mac OS, etc.	iv) Eg. Nucleus, Lynx OS,

② RTOS → real time operating system.

i) Reliability -

ii) predictability -

iii) Performance - fast કામ કરે

iv) Compactness - small & efficient કામ કરવો

v) Scalability :- capable of adding/deleting modular components

(3) Task → a simple subroutine.
a set of program instructions that are loaded in memory.

(4) Task states :-
i) Running or Executing
ii) Blocked (waited, dormant, delayed)
iii) Ready (pending, suspended)

Task global simply => code + Data + State.

(5) TCB (Task Control Block)
Task state is stored in TCB.

TCB मी काय असते ?

ID
priority
status
Registers
saved PC
saved SP

(6) ~~जेव्हा~~ जेव्हा two tasks with same priority are ready?

→ If one task is running & another high priority task unblocks,

IPC - Inter process Communication

(6) process vs Threads

process	Thread
i) program in execution	i) function within executable
ii) Requires separate address space.	ii) shares same address space.
iii) IPC is expensive	iii) IPC is less expensive.
iv) process can own memory SoC	iv) It has shared memory

(7) kernel

→ most fundamental part of an OS upon which all other components depend.

→ responsible for management of tasks & communication betn tasks.

(8) Essential components of Real time kernel

→ Scheduler
→ Objects
→ Services

Scheduler - heart of kernel

i) preemptive scheduling - High-priority task preempts the low-priority task

ii) Non-preemptive scheduling - low priority task relinquishes the CPU

Khom

Difference

(9) Preemptive	Non-preemptive.
i) flexible so on	i) It is rigid
ii) process can be interrupted in bet"	ii) process can not be interrupted till it
iii) overheads so on	iii) no overhead.

(10) What is Reentrancy?

Function that can be interrupted at any time and resumed at a later time without loss of data.

(11) 3 rules to decide if a function is reentrant.

i) It uses all shared variables in an atomic way
ii) It does not call non-entrant functions.
iii) It does not use the hlw in a non-atomic way.

(12) Semaphore (imp)

new tool to deal with shared data problems.

Terms used in semaphore:

- get & give
- take & release
- pend & post
- P & V
- wait & signal

(13) Commonly used semaphore is Binary Semaphore

(14) What are operations of semaphore?
→ Initialize
→ wait
→ signal

(15) Two RTOS functions in Binary Semaphore.
→ TakeSemaphore
→ ReleaseSemaphore.

(16) Only one task can have a semaphore at a time.

(17) Types of semaphore.
→ Binary Semaphore
→ Counting "
→ Mutex "

(18) kernel Services
→ memory management → allocate & free memory
→ Timer functions → keep track of passage of time
→ Interrupt processing

Rule1. An interrupt routine must not call any RTOS func^n that might block the caller.

Rule 2. An interrupt routine must not call any RTOS func^n that might cause the RTOS to switch tasks.
unless the RTOS knows an interrupt routine.

Chap. 6

Embedded Software Development Tools

(1) What is cross Assemblers?

→ It converts the source program written in assembly language into the machine language of the processor.

(2) Cross Compilers:-

A cross compiler is a compiler capable of creating executable code for a platform other than the one which the compiler is running

(3) Locator

→ Locator ची काय गरज

It is used for linking & loading of the object programs.

(4) Debugger (Debugging tool)

→ It is a computer program that is used to test and debug other programs code.

The debugger may be.

(i) Source-level debugger (~~low level debugger~~)

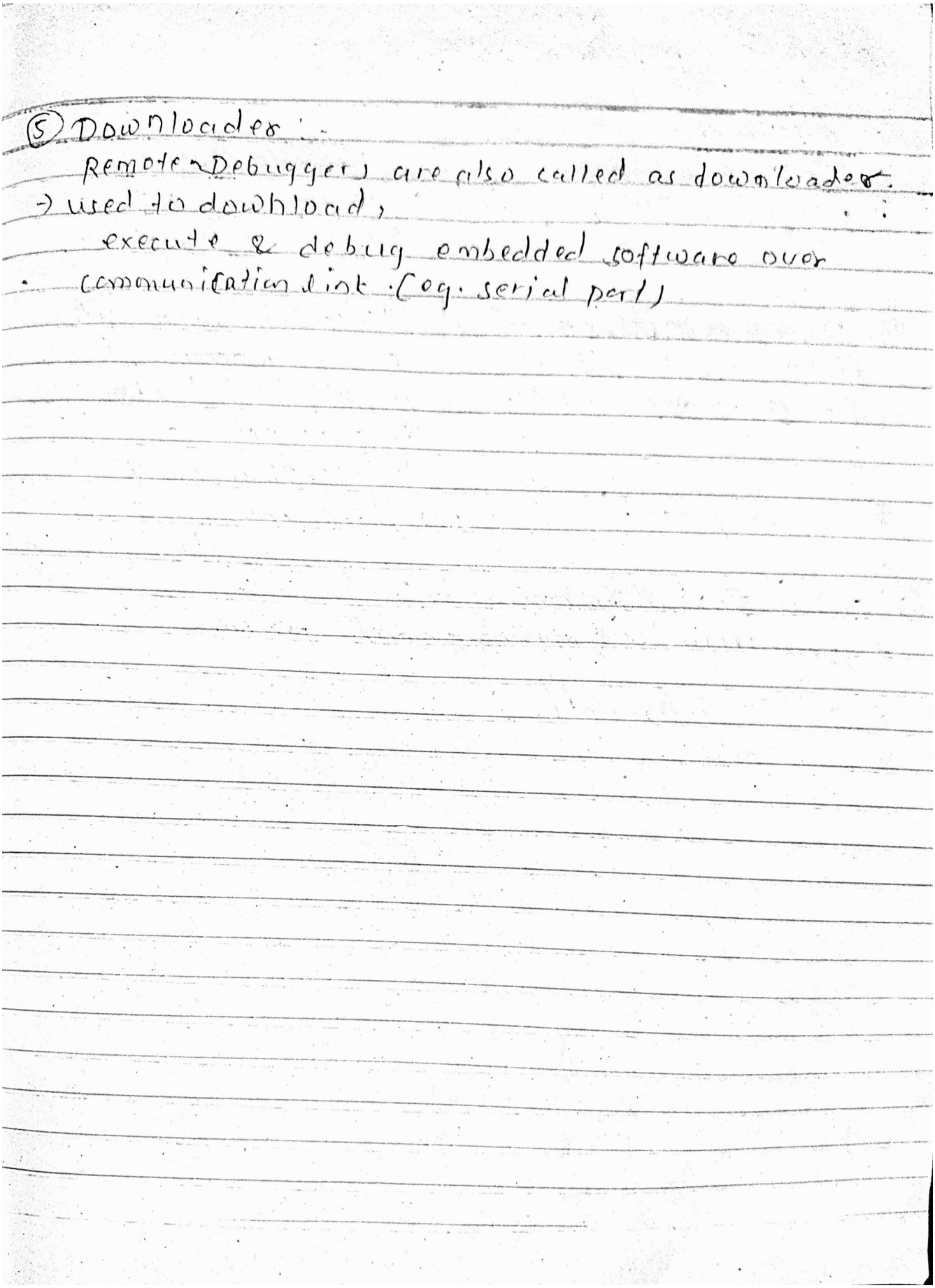
→ It can show the actual position in the original code.

(ii) Low-level debugger (machine-language debugger)

→ it shows that line in the program.

Embedded System viva question collection

Khom



(5) Downloader :-

Remote Debuggers are also called as downloader.

→ used to download, execute & debug embedded software over communication link (eg. serial port)

Khom

Chap. 7 8051 Micro controllers

① Difference :

Microprocessors	microcontrollers
i) Few bit handling instructions.	i) It has many bit handling instructions.
ii) It contains ALU, General purpose registers, stack pointers, program counter,	ii) It contains the circuitry of microprocessor, & in addition it has built in ROM, RAM, I/O devices, Timers/Counters, etc.
iii) low cost	iii) high cost

② Why 8051 microcontroller ?

→ very versatile featuring powerful processor.

- handles interrupts.
- low cost.

Chap. 8 VHDL → [Very High Speed Integrated Circuit Hardware Description Language]

(1) Why VHDL?

:) i) Shorter development time for electronic design
ii) simpler maintenance.
iii) Traditional: schematic design.

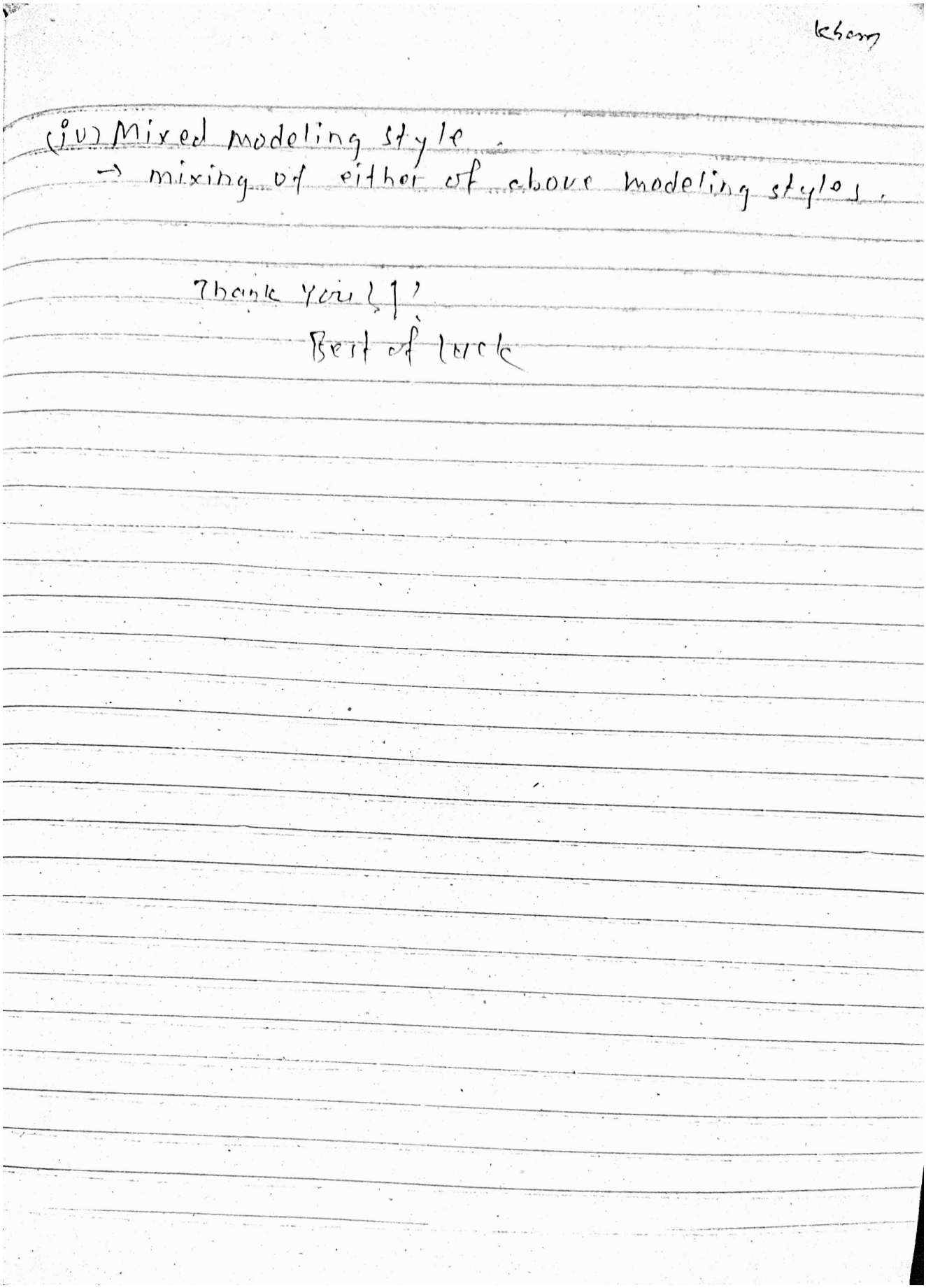
(2) Types Of VHDL

i) Behavioral modeling style
ii) Data flow " "
iii) structural " "
iv) mixed " "

(i) Behavioral modeling style:
specifies the behavior of an entity.
- sequential statements §6 Ep

(ii) Data flow.
Here, the flow of data through entity is expressed primarily using concurrent signal assignment statements.

(iii) Structural.
In this, an entity is described as a set of interconnected components.
It has two parts
→ component declaration.
→ component instantiation.



khom

(iv) Mixed modeling style.

→ mixing of either of above modeling styles.

Thank you !!

Best of luck