Q.2 a. What is a "market window" and why it is so important for products to reach the market early in this window?

Answer:

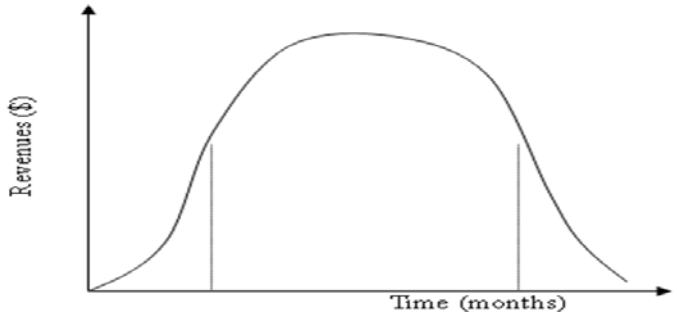


Fig: Market Window

- Market window
  - Period during which the product would have highest sales
- Time required to develop a product to the point it can be sold to customers
- Average time-to-market constraint is about 8 months
- Delays can be costly

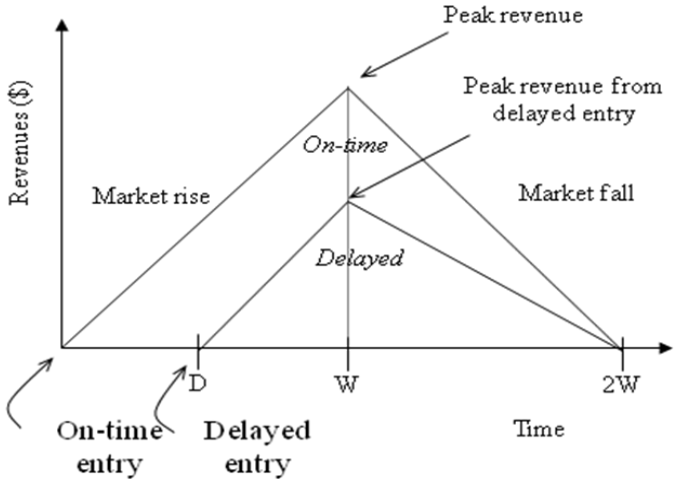


Fig: Revenue Model for computing revenue loss from delayed entry

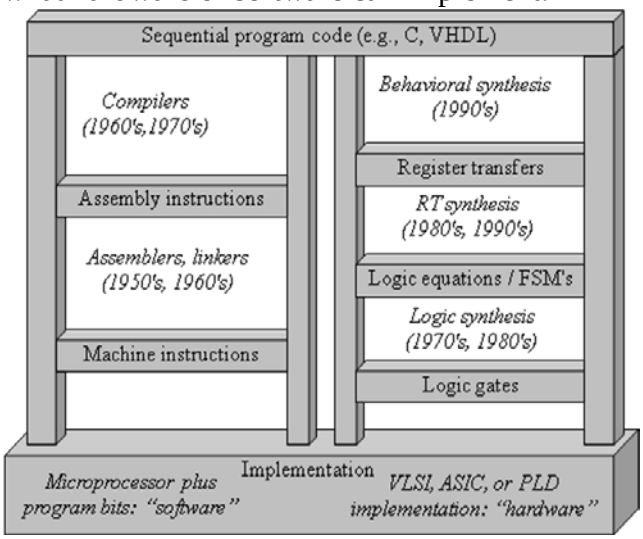
- Simplified revenue model
  - Product life = 2W, peak at W
  - Time of market entry defines a triangle, representing market penetration
  - Triangle area equals revenue
- Loss
  - The difference between the on-time and delayed triangle areas
- Area = 1/2 * base * height
  - On-time = 1/2 * 2W * W
  - Delayed = 1/2 * (W-D+W)*(W-D)

- Percentage revenue loss = $(D(3W-D)/2W^2)*100\%$

b. Define Moore's law. Explain co-design ladder in embedded system.

Answer:

- In the past:
  - Hardware and software design technologies were very different
  - Recent maturation of synthesis enables a unified view of hardware and software
- Hardware/software "co-design"
- The choice of hardware versus software for a particular function is simply a tradeoff among various design metrics, like performance, power, size, NRE cost, and especially flexibility; there is no fundamental difference between what hardware or software can implement.

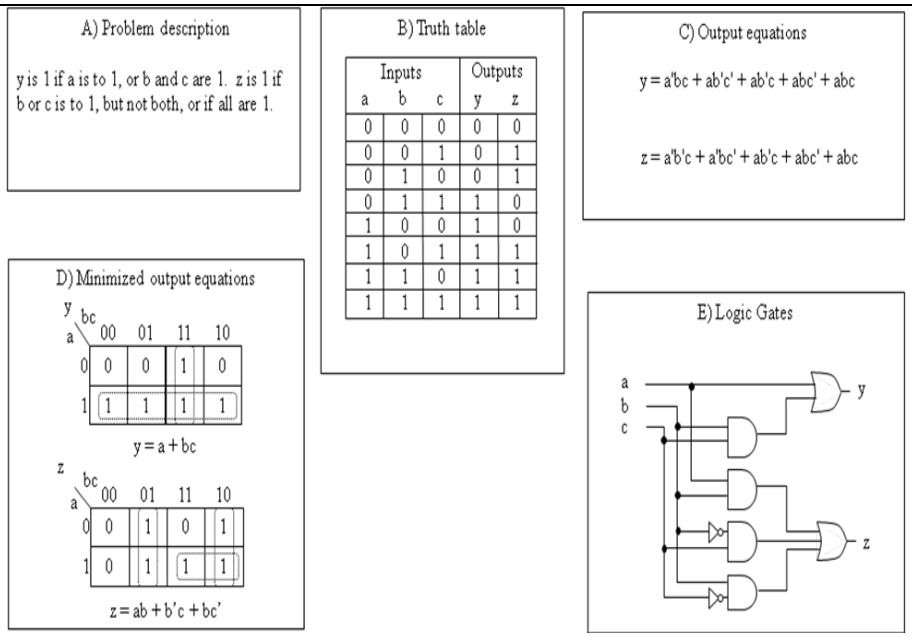


Q.3 a. Design a combinational circuit for a problem "y is 1 if a is 1, or b and c are 1, z is 1 if b or c is 1, but not both".

Answer:

Design a combinational circuit for a problem "y is 1 if a is 1, or b and c are 1z is 1 if b or c is 1, but not both". (10)

Answer:
Combinational Logic Design:



b. Explain the different methods of Optimizing the FSMD.

Answer:

- Areas of possible improvements
  - merge states
    - states with constants on transitions can be eliminated, transition taken is already known
    - states with independent operations can be merged
  - separate states
    - states which require complex operations (a*b*c*d) can be broken into smaller states to reduce hardware size
  - scheduling

Q.4 a. Why composing of larger memory is required from smaller memory parts? Explain, how you will approach this method?

Answer:

An embedded system designer is often faced with the situation of needing a particular-sized memory (ROM or RAM), but having readily available memories of a different size.

Explanation: (Refer Text1- page no: 123- 124)

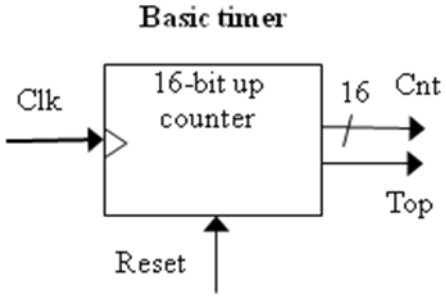
b. Draw and explain the general purpose processor architecture.

Answer: Refer Page No. 77 from Text Book

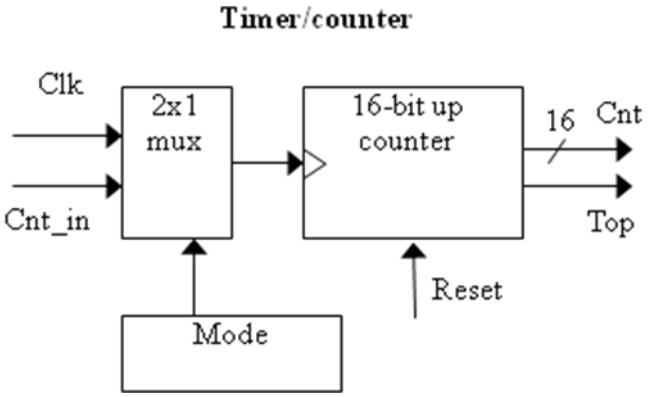
Q.5 a. Explain the features of Timers, Counters and Watchdog Timers.

Answer:

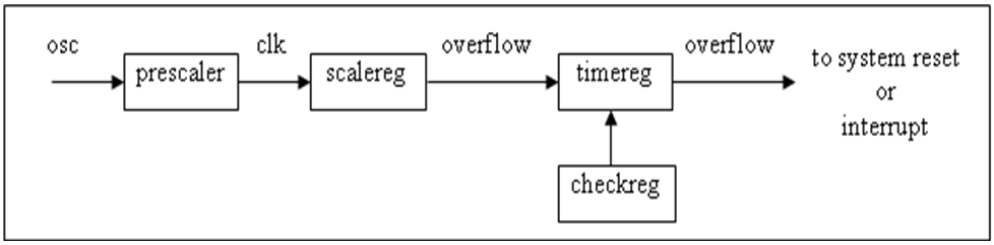
- Timer: measures time intervals
  - To generate timed output events
    - e.g., hold traffic light green for 10 s
  - To measure input events
    - e.g., measure a car's speed



- Counter: like a timer, but counts pulses on a general input signal rather than clock
  - e.g., count cars passing over a sensor
  - Can often configure device as either a timer or counter



- Watchdog timer
  - Must reset timer every X time unit, else timer generates a signal
  - Common use: detect failure, self-reset



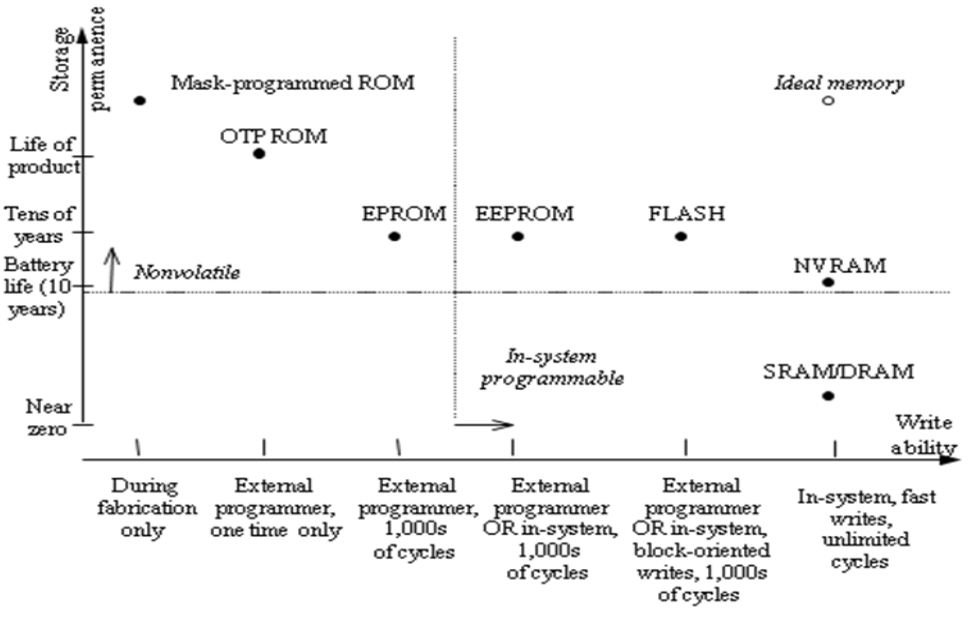
b. Given an analog input signal whose voltage is ranging from 0 to 15V, and an 8-bit digital encoding. Calculate the correct encoding of 5V using Successive Approximation.

Answer: Refer Pg No. 104-105 from Text Book-I

Q.6 a. Explain all memories write ability/ storage permanence.

Answer:

- Traditional ROM/RAM distinctions
  - ROM
    - read only, bits stored without power
  - RAM
    - read and write, lose stored bits without power
- Traditional distinctions blurred
  - Advanced ROMs can be written to
    - e.g., EEPROM
  - Advanced RAMs can hold bits without power
    - e.g., NVRAM
- Write ability
  - Manner and speed a memory can be written
- Storage permanence
  - ability of memory to hold stored bits after they are written

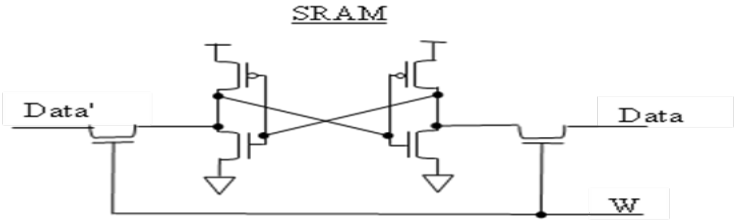


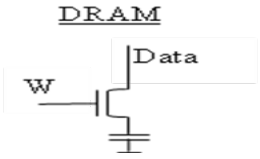
Write ability and storage permanence of memories, showing relative degrees along each axis (not to scale).

b. Draw the SRAM and DRAM circuit structures and list their main features.

Answer:

- SRAM: Static RAM
  - Memory cell uses flip-flop to store bit
  - Requires 6 transistors
  - Holds data as long as power supplied
- DRAM: Dynamic RAM
  - Memory cell uses MOS transistor and capacitor to store bit
  - More compact than SRAM
  - "Refresh" required due to capacitor leak
    - word's cells refreshed when read
  - Typical refresh rate 15.625 microsec.
  - Slower to access than SRAM





Q.7 a. Draw the timing diagram for a bus protocol that is handshaked non addressed and transfer 8 bits of data of over a 4 bit data bus.

Answer: Refer Page No. 176 from Text Book

b. Discuss the advantages and disadvantages of using memory-mapped I/O versus standard I/O.

Answer:

In bus-based I/O, there are two methods for a microprocessor to communicate with peripherals, known as memory – mapped I/O and standard I/O.

In memory mapped I/O, peripherals occupy specific addresses in the existing address space.

Example: Consider a bus with a 16-bit address. The lower 32K address may correspond to memory addresses, while the upper 32K may correspond to I/O

addresses.

In standard I/O (also known as I/O – mapped I/O) the bus includes an additional pin which we label M/IO, to indicate whether the access is to memory or to a peripheral.

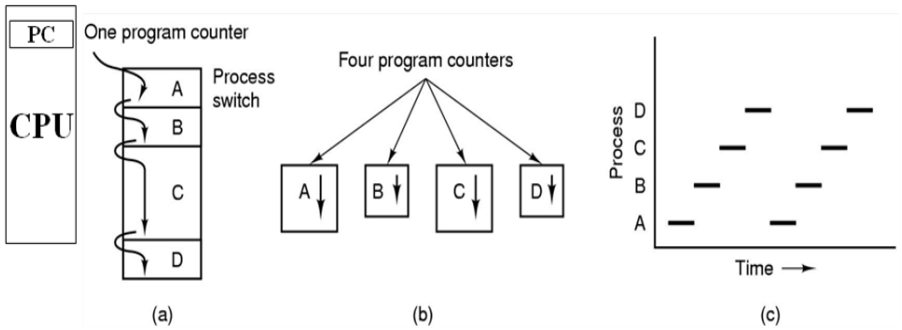
The advantage of memory-mapped I/O is that the microprocessor need not include special instructions for communicating with peripherals. In contrast, if the microprocessor uses a standard I/O, the microprocessor requires special instructions for reading and writing peripherals. The advantages of standard I/O include no loss of memory addresses to the use as I/O addresses, and potentially simpler address decoding logic peripherals.

Q.8 a. Explain the Process and Task concepts in RTOS.

Answer:

- An operating system executes a variety of programs:
  - Batch system – jobs
  - Time-shared systems – user programs or tasks
- Similar terms *job*, *process*, *task (ES)* almost interchangeably
- Process – a program in execution; process execution must progress in sequential fashion
- A process includes:
  - program counter
  - stack
  - data section

Example of Processes: The Process Model



- Multiprogramming of four programs
- Conceptual model of 4 independent, sequential processes
- Only one program active at any instant

b. Explain in brief, comparison of the methods for Inter-task communication.

Answer: Refer Page No. 192 from Text Book-II

Q.9 a. Draw the state diagram for automatic chocolate vending machine (AVCM) tasks.

Answer: Refer Page No. 518 from Text Book-III.

b. Draw and explain block diagram of AVCM hardware including microcontroller.

Answer: Refer Page No. 518 from Text Book-III

TEXT BOOKS

I. Embedded System Design, A Unified Hardware/Software Introduction, Frank Vahid / Tony Givargis, 2006 reprint, John Wiley Student Edition.

II. An Embedded Software Primer, David .E. Simon, Fourth Impression 2007, Pearson Education.

III. Embedded Systems, Raj Kamal, 13th reprint 2007, Tata-McGrawHill Publications.