Q.2 a. Define Embedded System. Explain the performance design metric of an embedded system.

Answer:

- Embedded computing systems
  - Computing systems embedded within electronic devices
  - Hard to define. Nearly any computing system other than a desktop computer

The performance design metric:

- Widely-used measure of system, widely-abused
  - Clock frequency, instructions per second – not good measures
  - Digital camera example – a user cares about how fast it processes images, not clock speed or instructions per second
- Latency (response time)
  - Time between task start and end
  - e.g., Camera's A and B process images in 0.25 seconds
- Throughput
  - Tasks per second, e.g. Camera A processes 4 images per second
  - Throughput can be more than latency seems to imply due to concurrency, e.g. Camera B may process 8 images per second (by capturing a new image while previous image is being stored).
- *Speedup* of B over S = B's performance / A's performance
  - Throughput speedup = 8/4 = 2

b. Compare Full Custom Design, Semi Custom Design and PLDs.

Answer:

Full-custom:

- All layers are optimized for an embedded system's particular digital implementation
  - Placing transistors
  - Sizing transistors
  - Routing wires
- Benefits
  - Excellent performance, small size, low power
- Drawbacks
  - High NRE cost (e.g., $300k), long time-to-market
  -

Semi-custom:

- Lower layers are fully or partially built
  - Designers are left with routing of wires and maybe placing some blocks
- Benefits
  - Good performance, good size, less NRE cost than a full-custom implementation (perhaps $10k to $100k)

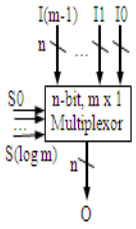
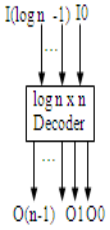
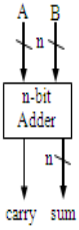
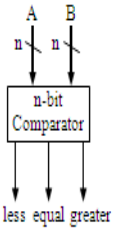
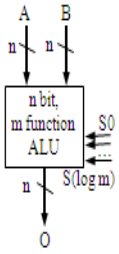
• Drawbacks
  – Still require weeks to months to develop

PLD (Programmable Logic Device):
• All layers already exist
  – Designers can purchase an IC
  – Connections on the IC are either created or destroyed to implement desired functionality
  – Field-Programmable Gate Array (FPGA) very popular
• Benefits
  – Low NRE costs, almost instant IC availability
• Drawbacks
  – Bigger, expensive (perhaps $30 per unit), power hungry, slower

Q.3 a. What is the use of RTL components in combinational circuits? Explain important RTL components which are used in combinational circuits design.

Answer:

One way to reduce the complexity is to use combinational components that are more powerful than logic gates. The below schematic shows several such combinational components, often called register – transfer or RT, level components.

				
O = I0 if S=0..00 I1 if S=0..01 … I(m-1) if S=1..11	O0 =1 if I=0..00 O1 =1 if I=0..01 … O(n-1) =1 if I=1..11	sum = A+B (first n bits) carry = (n+1)'th bit of A+B	less = 1 if A<B equal =1 if A=B greater=1 if A>B	O = A *op* B *op* determined by S.
	With enable input e → all O's are 0 if e=0	With carry-in input Ci→ sum = A + B + Ci		May have status outputs carry, zero, etc.

b. Define Optimization. Explain optimization opportunities in a single-purpose processor.

Answer:

- Optimization is the task of making design metric values the best possible
- Optimization opportunities
  - original program
  - FSMD
  - datapath
  - FSM
- Analyze program attributes and look for areas of possible improvement
  - number of computations
  - size of variable
  - time and space complexity
  - operations used
    - multiplication and division very expensive
- Optimizing the FSMD
  - 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
- Optimizing the datapath
  - Sharing of functional units
    - one-to-one mapping, as done previously, is not necessary
    - if same operation occurs in different states, they can share a single functional unit
  - Multi-functional units
  - ALUs support a variety of operations, it can be shared among operations occurring in different states
- Optimizing the FSM
  - State encoding
    - task of assigning a unique bit pattern to each state in an FSM
    - size of state register and combinational logic vary
    - can be treated as an ordering problem
  - State minimization
    - task of merging equivalent states into a single state
  - state equivalent if for all possible input combinations the two states generate the same outputs and transitions to the next same state

Q.4 a. Define ASIP. Explain how it is different from general-purpose processors with one example.

Answer:

- ASIPs – targeted to a particular domain
  - Contain architectural features specific to that domain
    - e.g., embedded control, digital signal processing, video processing, network processing, telecommunications, etc.
  - Still programmable
- General-purpose processors
  - Sometimes too general to be effective in demanding application
    - e.g., video processing – requires huge video buffers and operations on large arrays of data, inefficient on a GPP
  - But single-purpose processor has high NRE, not programmable

A Common ASIP: Microcontroller:

- For embedded control applications
  - Reading sensors, setting actuators
  - Mostly dealing with events (bits): data is present, but not in huge amounts
  - e.g., VCR, disk drive, digital camera (assuming SPP for image compression), washing machine, microwave oven
- Microcontroller features
  - On-chip peripherals
    - Timers, analog-digital converters, serial communication, etc.
    - Tightly integrated for programmer, typically part of register space
  - On-chip program and data memory
  - Direct programmer access to many of the chip's pins
  - Specialized instructions for bit-manipulation and other low-level operations

b. Discuss the technical aspects of selecting a microprocessor for use in an embedded system.

Answer:

- Issues
  - Technical: speed, power, size, cost
  - Other: development environment, prior expertise, licensing, etc.
- Speed: how evaluate a processor's speed?
  - Clock speed – but instructions per cycle may differ
  - Instructions per second – but work per instr. may differ
  - Dhrystone: Synthetic benchmark, developed in 1984. Dhrystones/sec.

- MIPS: 1 MIPS = 1757 Dhrystones per second (based on Digital's VAX 11/780). A.k.a. Dhrystone MIPS. Commonly used today.
  - So, 750 MIPS = 750*1757 = 1,317,750 Dhrystones per second

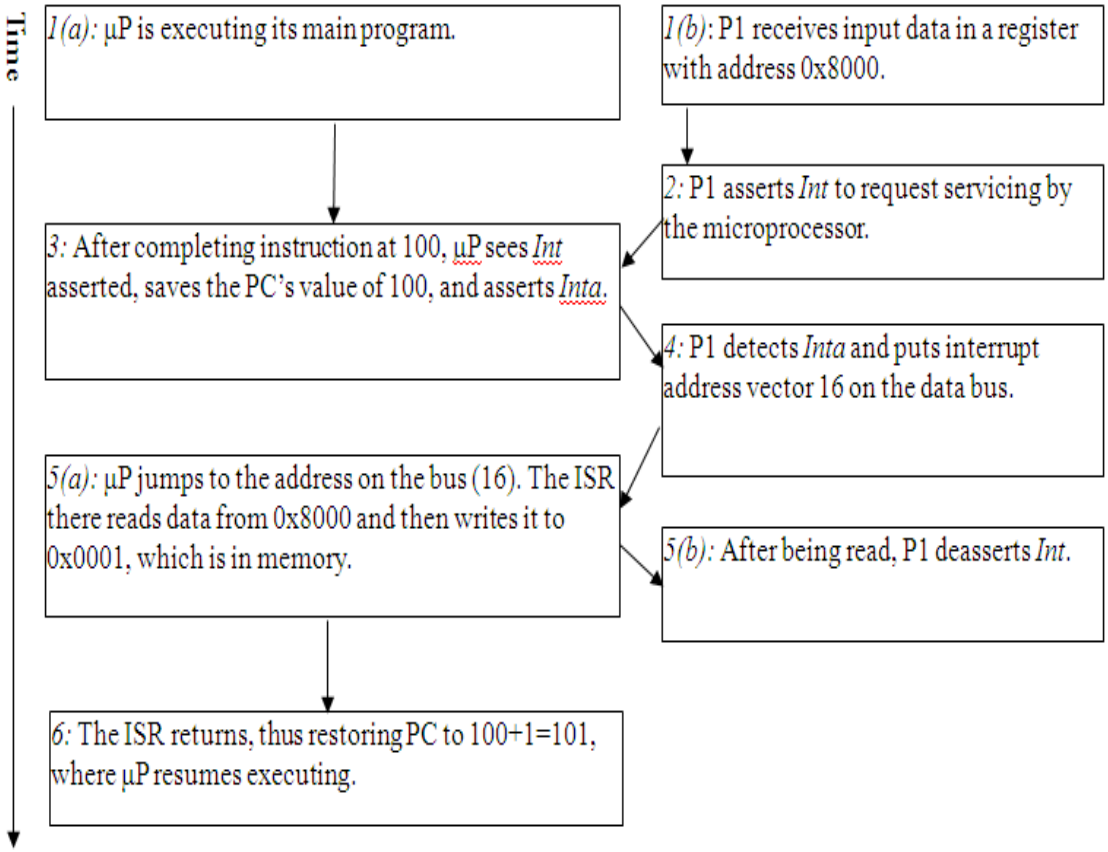
– SPEC: set of more realistic benchmarks, but oriented to desktops
– EEMBC – EDN Embedded Benchmark Consortium.
- Suites of benchmarks: automotive, consumer electronics, networking, office automation, telecommunications

Q.5 a. What is a timer? How does a counter perform:
(i) Timer function
(ii) Prefixed time initiated event generation
(iii) Time capture function

Answer: Page no 84 of Text Book

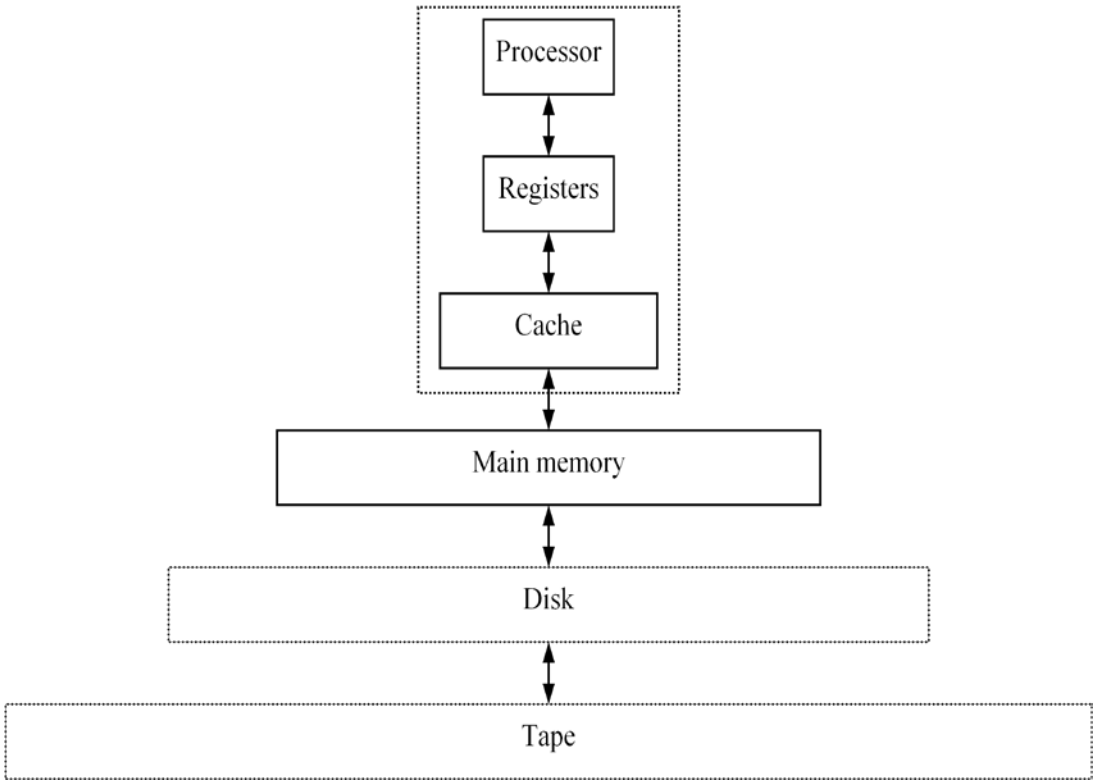
Q.6 a. Discuss all the steps used in peripheral to memory transfer without DMA, using vectored interrupt.

Answer:



b. Draw the block diagram of memory hierarchy structure used in an embedded system.

Answer:



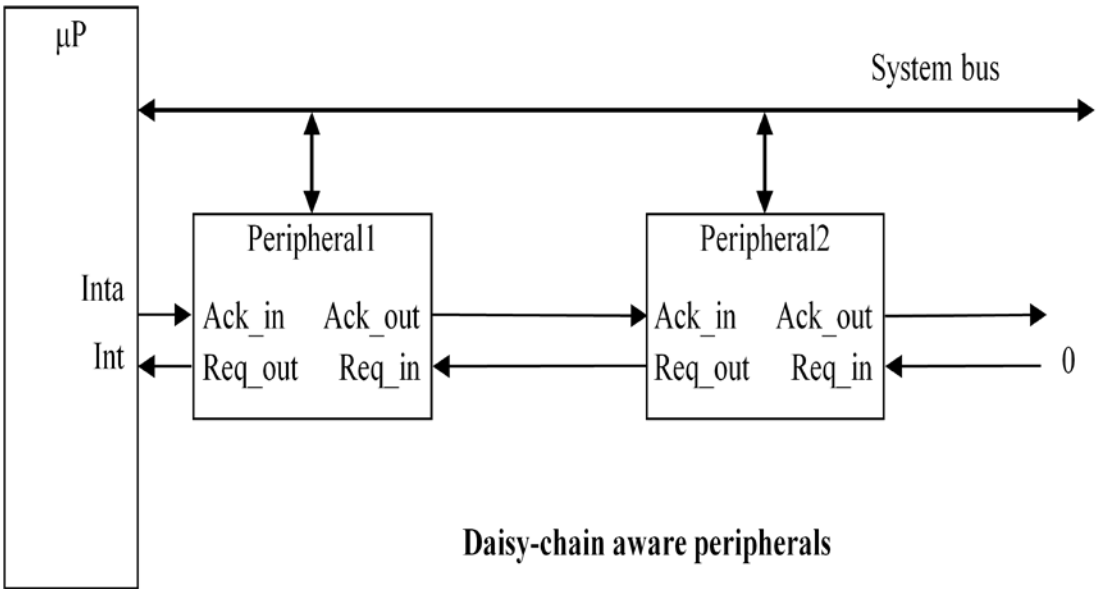
Q7 a. Explain the function of scheduler in detail.

Answer: Page no 140 of Text Book

b. Explain Daisy Chain arbitration and Network oriented arbitration.

Answer:

- Arbitration done by peripherals
  - Built into peripheral or external logic added
    - *req* input and *ack* output added to each peripheral
- Peripherals connected to each other in daisy-chain manner
  - One peripheral connected to resource, all others connected "upstream"
  - Peripheral's *req* flows "downstream" to resource, resource's *ack* flows "upstream" to requesting peripheral
  - Closest peripheral has highest priority



Network-oriented arbitration:

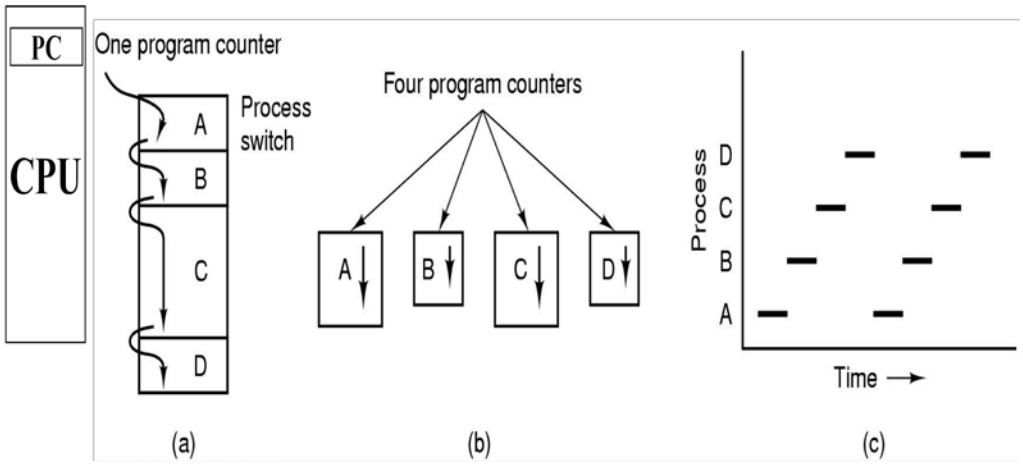
- When multiple microprocessors share a bus (sometimes called a network)
  - Arbitration typically built into bus protocol
  - Separate processors may try to write simultaneously causing collisions
    - Data must be resent
    - Don't want to start sending again at same time
      - statistical methods can be used to reduce chances
- Typically used for connecting multiple distant chips
  - Trend – use to connect multiple on-chip processors

Q8. 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: Page no 192 of Text Book

Q9 a. Draw the software architecture of an automatic chocolate vending machine (ACVM).

Answer: Page no 519 of Text Book

b. Draw and explain state diagram for ACVM tasks.

Answer: Page no 518 of Text Book

Text Books

1. **Embedded system design, A Unified Hardware/Software Introduction, Frank Vahid/ Tony Givargis, 2006 reprint, John Wiley student Edition**

2. **An Embedded Software primer, David E. Simon, 4th Impression 2007, Pearson Education**

3. **Embedded Systems, RajKamal, 13th reprint 2007, Tata-McGraw Hill Publications**