Q.1 a. Summarize the characteristics of Embedded Computing Applications.

Answer:

Embedded computing systems have to provide sophisticated functionality: **Complex algorithms:** The operations performed by the microprocessor may be very sophisticated. For example, the microprocessor that controls an automobile engine must perform complicated filtering functions to optimize the performance of the car while minimizing pollution and fuel utilization.

User interface: Microprocessors are frequently used to control complex user interfaces that may include multiple menus and many options. The moving maps in Global Positioning System (GPS) navigation are good examples of sophisticated user interfaces.

b. Define Supervisor Mode, Exceptions, and Traps.

Answer:

Complex systems are often implemented as several programs that communicate with each other. These programs may run under the command of an operating system. It may be desirable to provide hardware checks to ensure that the programs do not interfere with each other—for example, by erroneously writing into a segment of memory used by another program. Software debugging is important but can leave some problems in a running system; hardware checks ensure an additional level of safety. In such cases it is often useful to have a supervisor mode provided by the CPU.

An exception is an internally detected error. A simple example is division by zero.

A trap, also known as a software interrupt, is an instruction that explicitly generates an exception condition. The most common use of a trap is to enter supervisor mode

c. List the main features of Microcontrollers.

Answer:

Microcontroller features

1. On-chip peripherals: Timers, analog-digital converters, serial communication, etc.
2. Tightly integrated for programmer, typically part of register space
3. On-chip program and data memory
4. Direct programmer access to many of the chip's pins
5. Specialized instructions for bit-manipulation and other low-level operations

d. How does handshake ensure that when two devices want to communicate, one is ready to transmit and other is ready to receive?

Answer:

The handshake uses a pair of wires dedicated to the handshake: enq (meaning enquiry) and ack (meaning acknowledge). Extra wires are used for the data transmitted during the handshake. The four cycles are described below.

1. Device 1 raises its output to signal an enquiry, which tells device 2 that it should get ready to listen for data.
2. When device 2 is ready to receive, it raises its output to signal an acknowledgment. At this point, devices 1 and 2 can transmit or receive.
3. Once the data transfer is complete, device 2 lowers its output, signalling that it has received the data.
4. After seeing that ack has been released, device 1 lowers its output.

e. List the recognizing features of a compiler to optimize program.

Answer:

1. Expression Simplification
2. Dead Code Elimination
3. Procedure In-lining
4. Loop Transformations
5. Register Allocation
6. Scheduling
7. Instruction Selection
8. Understanding and Using Your Compiler

f. Why would anyone build a distributed embedded system?

Answer:

Building an embedded system with several processing elements (PEs) talking over a network is definitely more complicated than using a single large microprocessor to perform the same tasks. Distributed systems are necessary because the devices that the PEs communicates with are physically separated. If the deadlines for processing the data are short, it may be more cost-effective to put the PEs where the data are located rather than build a higher-speed network to carry the data to a distant, fast PE.

An important advantage of a distributed system with several CPUs is that one part of the system can be used to help diagnose problems in another part. If you have several CPUs in the system, you can use one to generate inputs for another and to watch its output.

g. What are the different types of power management? Explain the need of it.

Answer:

There are two types of power management features provided by CPUs. A static power management mechanism is invoked by the user but does not otherwise depend on CPU activities. An example of a static mechanism is a power down mode intended to save energy. This mode provides a high-level way to reduce unnecessary power consumption. The mode is typically entered with an instruction. If the mode stops the interpretation of instructions, then it clearly cannot be exited by execution of another instruction. Power-down modes typically end upon receipt of an interrupt or other event. A dynamic power management mechanism takes actions to control power based upon the dynamic activity in the CPU. For example, the CPU may turn off certain sections of the CPU when the instructions being executed do not need them.

Q.2 a. Explain the Embedded System Design Challenges for optimizing design metrics.

Answer:

Obvious design goal: Construct an implementation with desired functionality. Key design challenge: Simultaneously optimize numerous design metrics Design metric: A measurable feature of a system's implementation optimizing design metrics is a key challenge

Common metrics:
Unit cost: the monetary cost of manufacturing each copy of the system, excluding NRE cost
NRE cost (Non-Recurring Engineering cost): The one-time monetary cost of designing the system
Size: the physical space required by the system
Performance: the execution time or throughput of the system
Power: the amount of power consumed by the system
Flexibility: the ability to change the functionality of the system without incurring heavy NRE cost

b. Explain the system parameters used for performance design metric.

Answer:

Widely-used measure of system, widely-abused
1. 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
2. Latency (response time) Time between task start and end
e.g., Camera's A and B process images in 0.25 seconds
3. 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).
4. Speedup of B over S = B's performance / A's performance
Throughput speedup = 8/4 = 2

Q.3 a. What are ASIP's? Explain the features of DSP & SOC ASIP's.

Answer:

Application-Specific Instruction-Set Processors (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 it's a programmable.

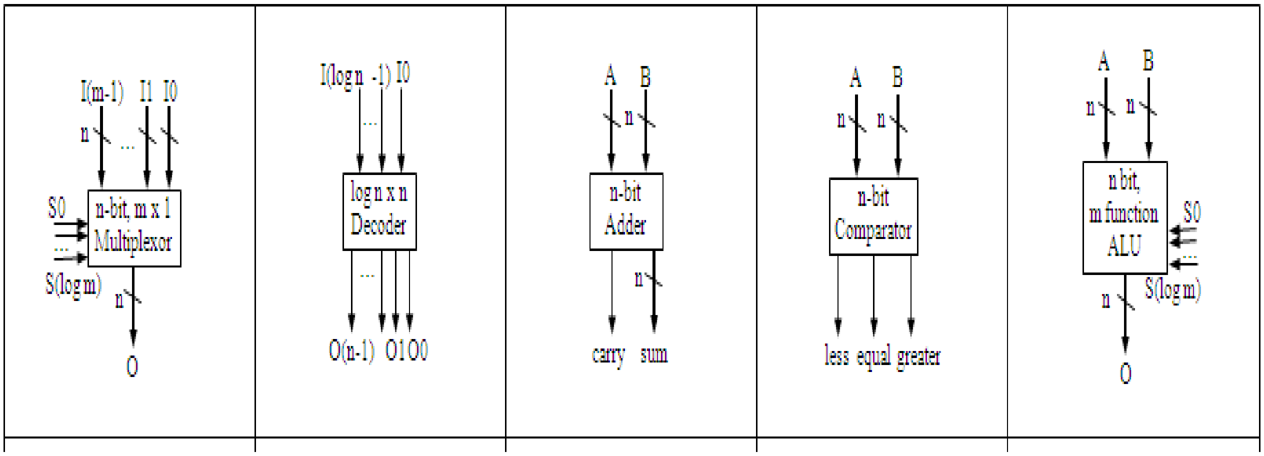
- For signal processing applications
  - Large amounts of digitized data, often streaming
  - Data transformations must be applied fast
  - e.g., cell-phone voice filter, digital TV, music synthesizer
- DSP features
  - Several instruction execution units
  - Multiple-accumulate single-cycle instruction, other instrs.
  - Efficient vector operations – e.g., add two arrays
    - Vector ALUs, loop buffers, etc.

SOC can be defined as integration of a complete system onto a single IC. Very large transistor counts on a single IC. The following are the characteristics of SOC's. Mixed technologies on the same chip - Digital, memory, analog, FPGA. Hardware and software Multiple clock frequencies Hierarchical design with embedded reusable IP cores

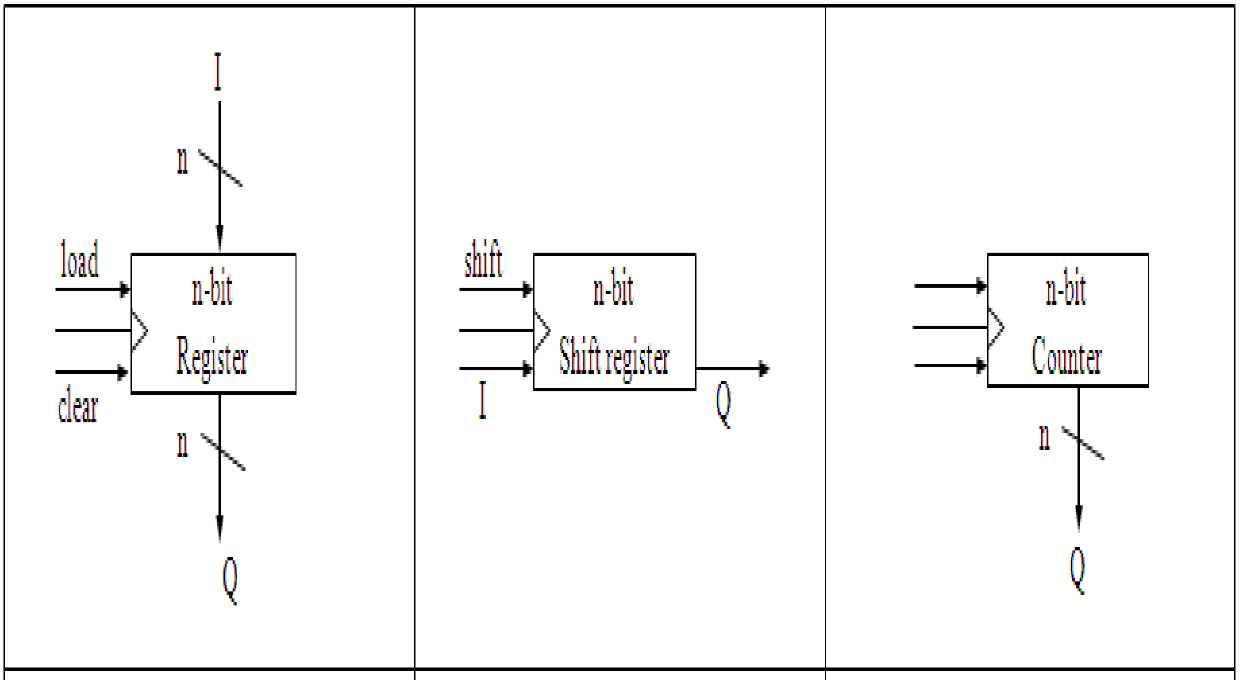
b. Draw and explain the Combinational & Sequential Components which can be used in embedded hardware system design.

Answer:

Combinational Components:

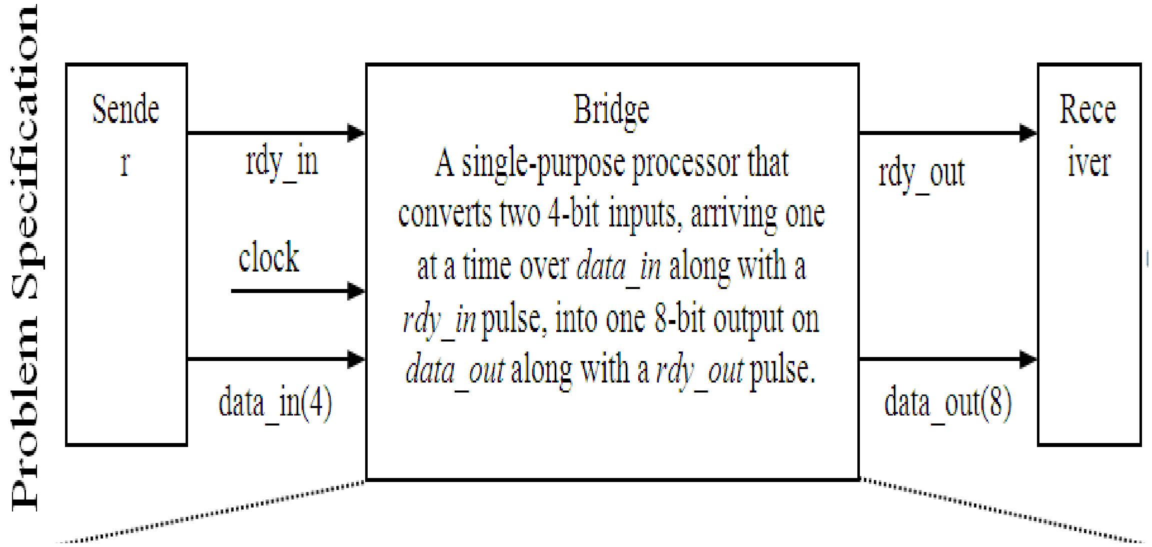


Sequential Components:



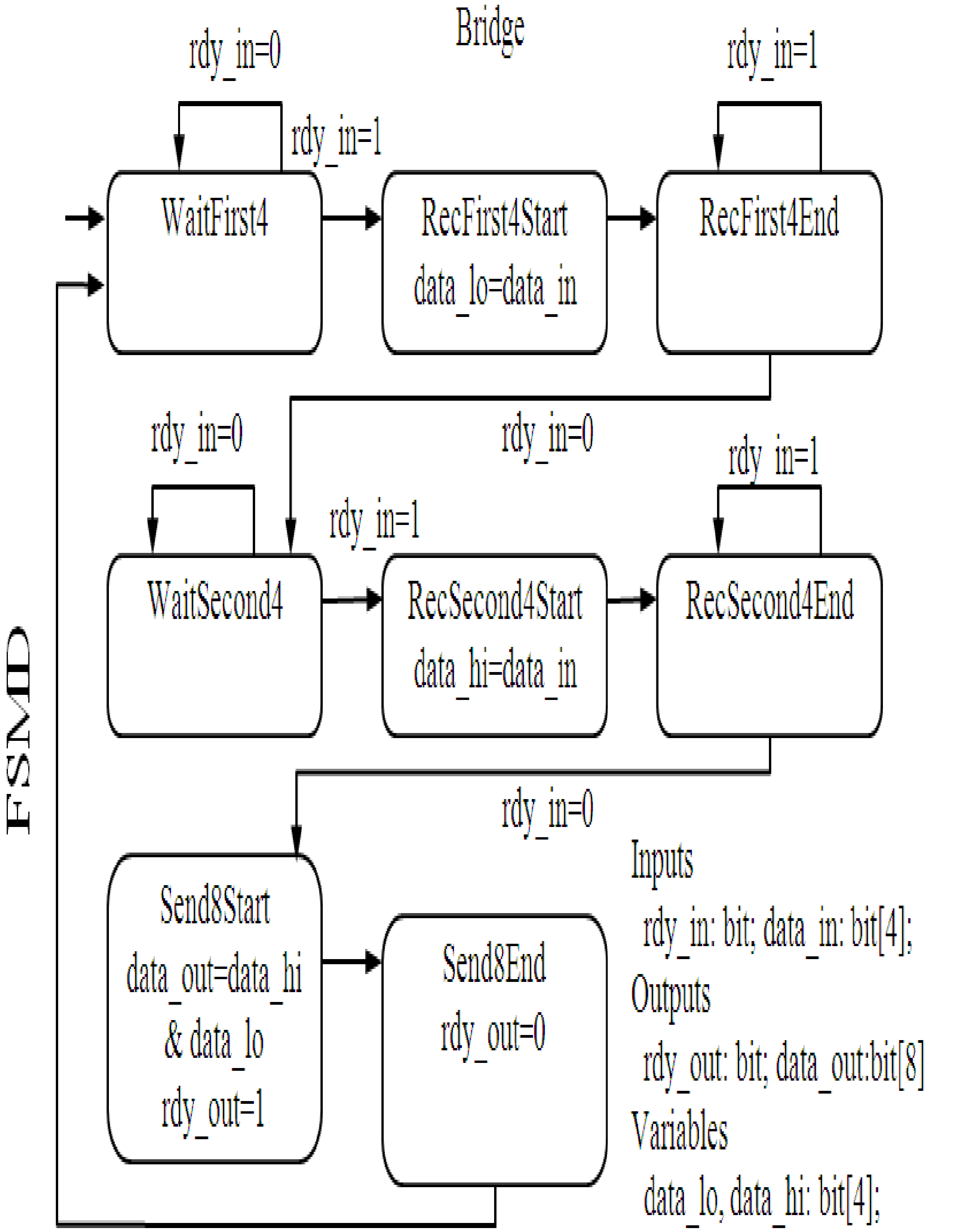
Q.4a. Design an RTL custom single-purpose processor bus bridge that converts 4-bit bus to 8-bit bus.

Answer:

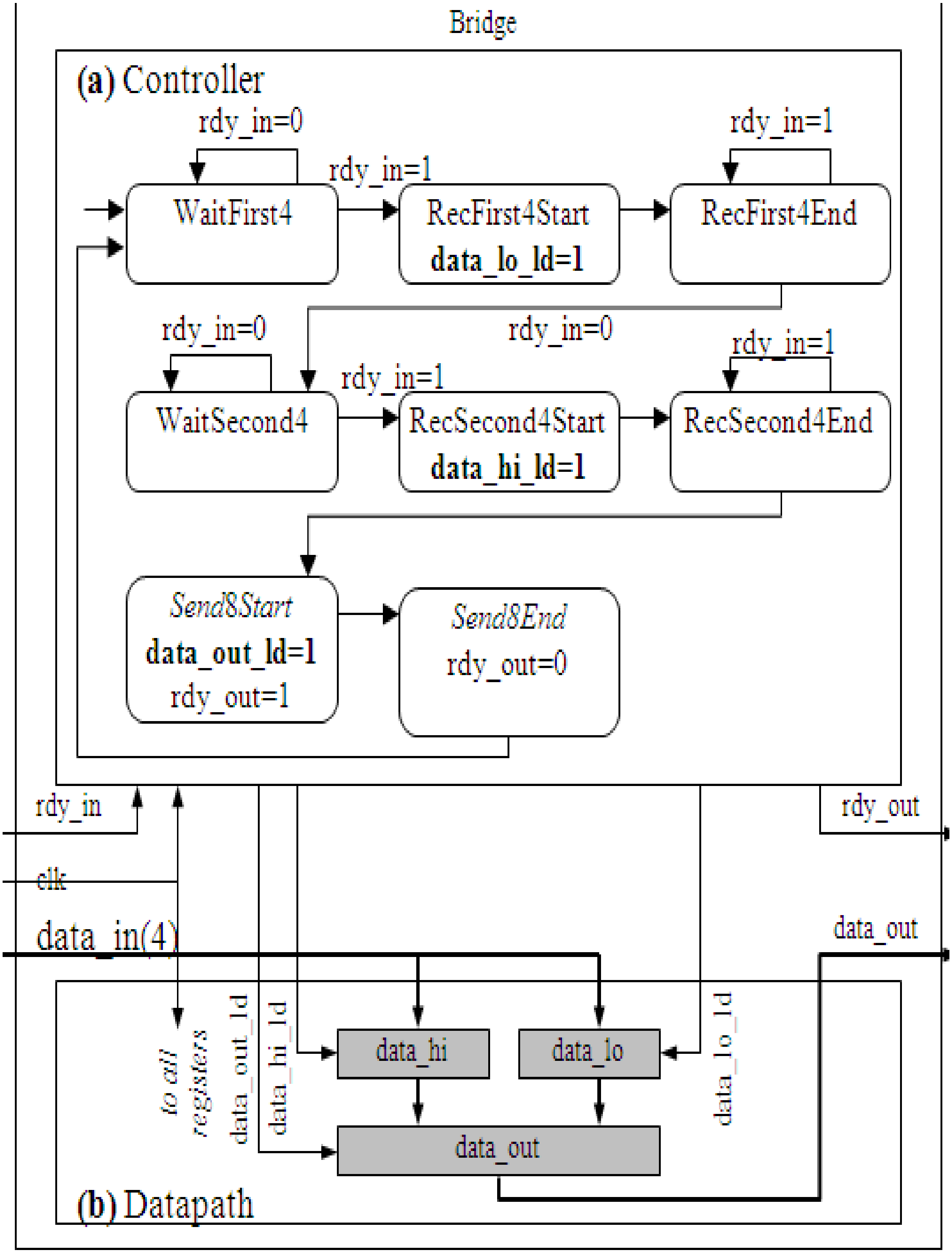


A single-purpose processor that converts two 4-bit inputs, arriving one at a time over *data_in* along with a *rdy_in* pulse, into one 8-bit output on *data_out* along with a *rdy_out* pulse.

FSMD :



Control Path & Data Path Design:


Bridge
(a) Controller
rdy_in=0
rdy_in=1
rdy_in=1
WaitFirst4
RecFirst4Start
data_lo_ld=1
RecFirst4End
rdy_in=0
rdy_in=0
rdy_in=1
rdy_in=1
WaitSecond4
RecSecond4Start
data_hi_ld=1
RecSecond4End
Send8Start
data_out_ld=1
rdy_out=1
Send8End
rdy_out=0
rdy_in
clk
data_in(4)
rdy_out
data_out
to all registers
data_out_ld
data_hi_ld
data_hi
data_lo
data_lo_ld
data_out
(b) Datapath

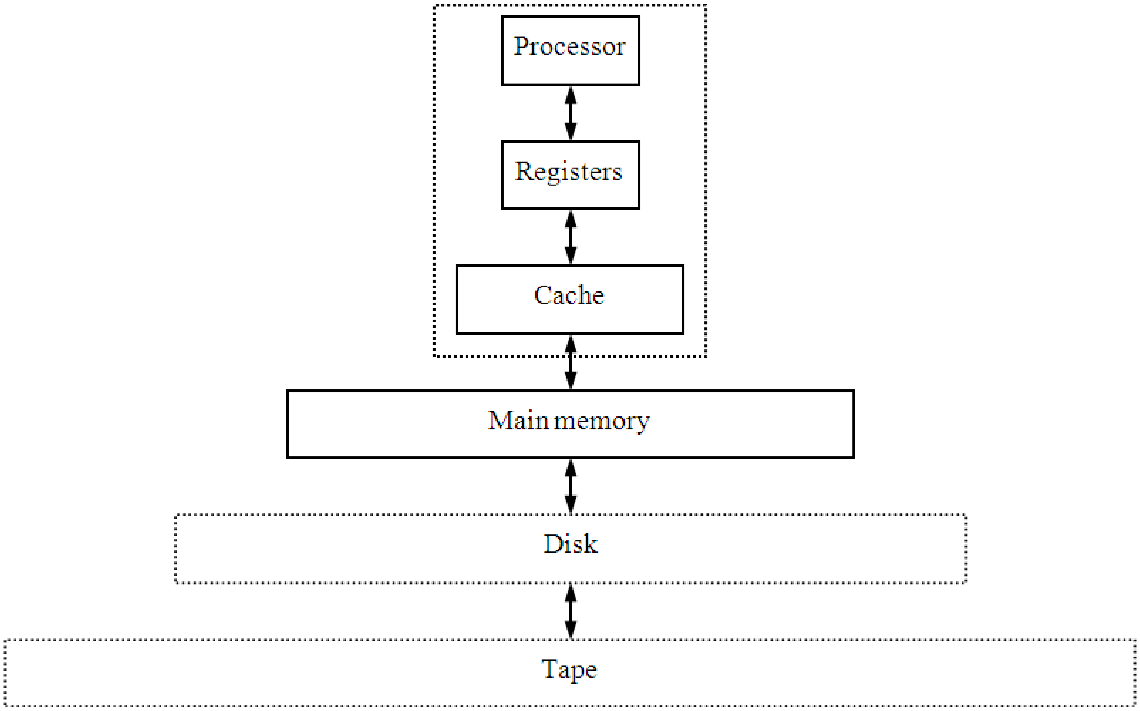
b. Explain the characteristics and features of a One-time programmable ROM

Answer:

- Connections “programmed” after manufacture by user
  - user provides file of desired contents of ROM
  - file input to machine called ROM programmer
  - each programmable connection is a fuse
  - ROM programmer blows fuses where connections should not exist
- Very low write ability
  - typically written only once and requires ROM programmer device
- Very high storage permanence
  - bits don’t change unless reconnected to programmer and more fuses blown
- Commonly used in final products
  - cheaper, harder to inadvertently modify

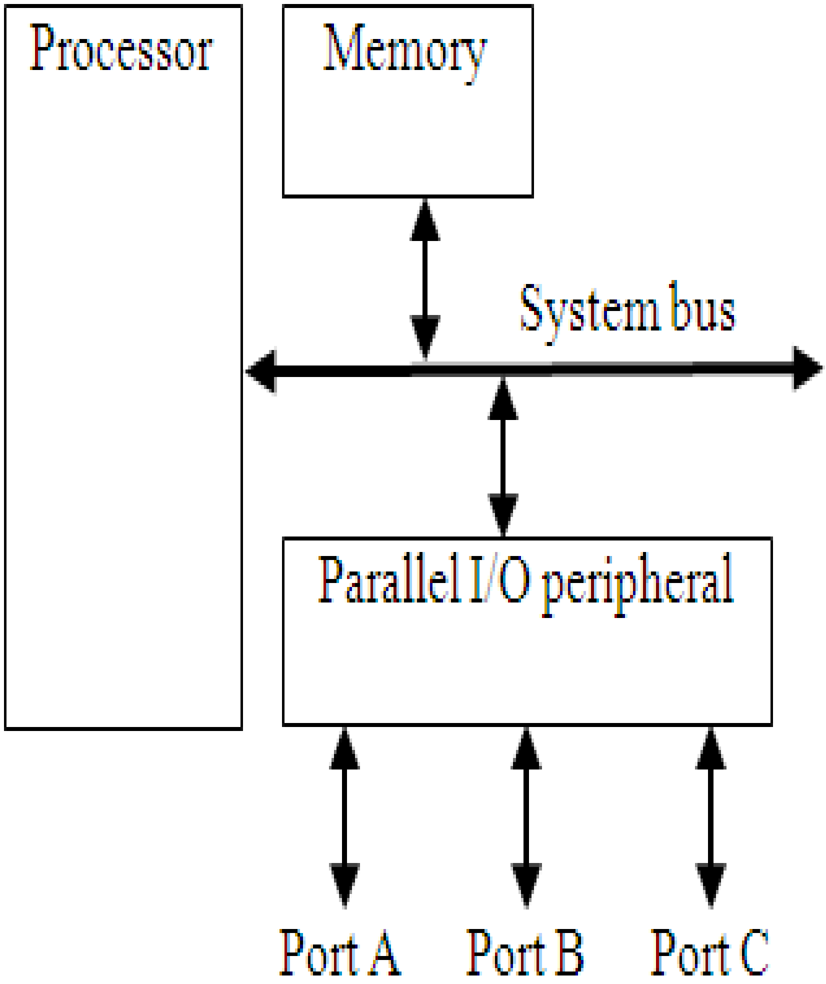
Q.5 a. Draw and explain the block diagram of a Memory hierarchy.

Answer:



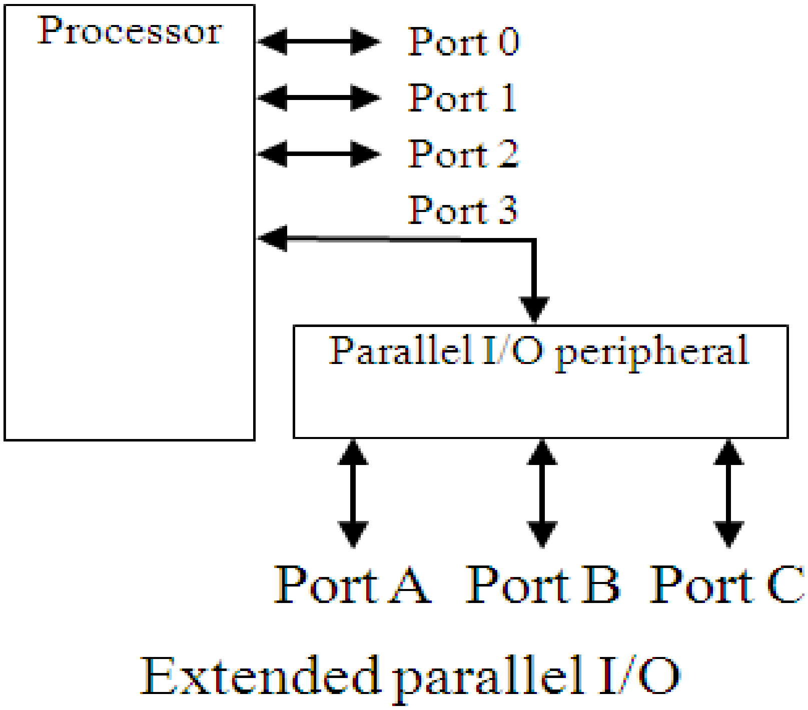
b. Explain with neat sketches Parallel I/O peripheral and Extended parallel I/O.

Answer:



Adding parallel I/O to a bus-based I/O processor

- Parallel I/O peripheral
  - When processor only supports bus-based I/O but parallel I/O needed
  - Each port on peripheral connected to a register within peripheral that is read/written by the processor



- Extended parallel I/O
  - When processor supports port-based I/O but more ports needed
  - One or more processor ports interface with parallel I/O peripheral extending total number of ports available for I/O
  - e.g., extending 4 ports to 6 ports in figure

Q.6

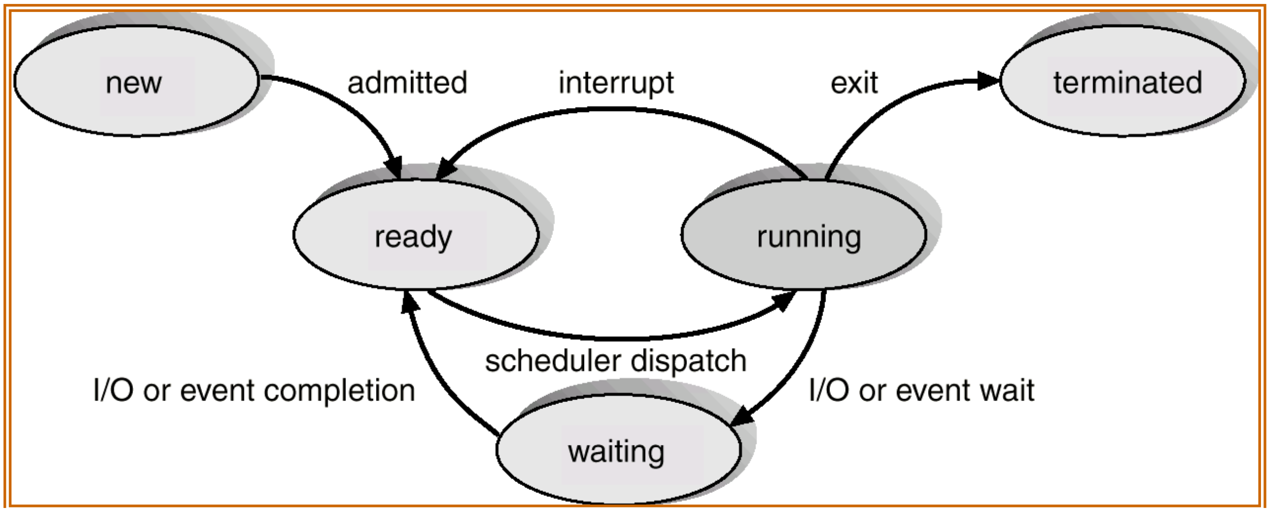
b. Explain the concept of Process/Task State in Embedded Operating System. Also draw diagram of Process State.

Answer:

Process/Task State:

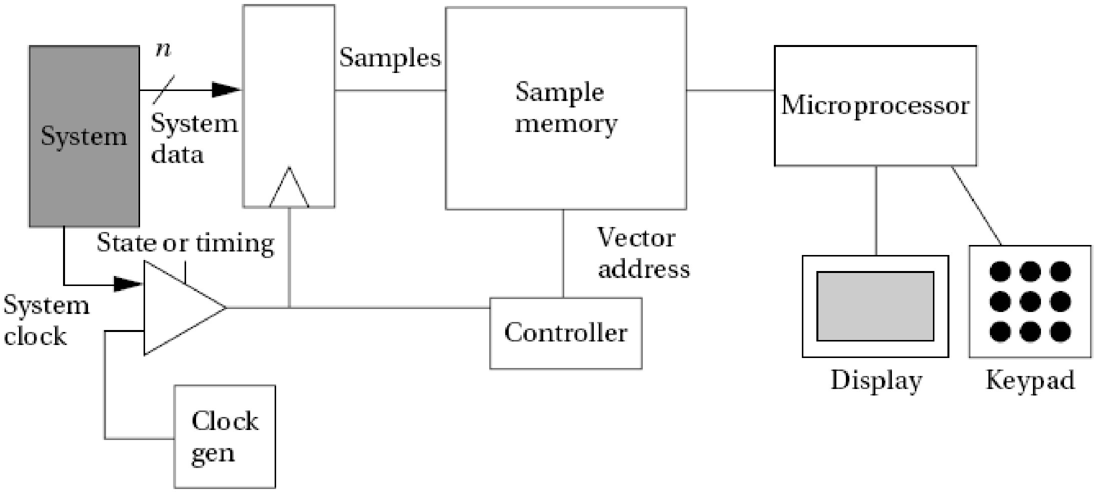
- **Bottom line**: keep track of which process/task runs (has the CPU)
- As a process executes, it changes *state (in some states, process/task does need CPU)*
  - **new**: The process is being created
  - **running**: Instructions are being executed
  - **waiting**: The process is waiting for some event to occur
  - **ready**: The process is waiting to be assigned to a process
  - **terminated**: The process has finished execution

Diagram of Process State:



Q.7 Draw and Explain the Architecture of a Logic Analyzer.

Answer:



Text Books

1. **Wayne Wolf, Computers as Components: Principles of Embedded Computing System Design, 2nd Edition, Morgan Kaufmann Publishers, 2008**

2. **Frank Vahid and Tony Givargis, Embedded System Design: A unified Hardware/Software Introduction, John Wiley & Sons, 2002.**