Computer Architecture
Homework 3
Theme: Buses

All questions are of equal weight. Show your work to receive credit.

---

1. At what point during an instruction cycle an external interrupt is recognized by the CPU? When more than one interrupt occurs, what approaches can you use to service them? What approaches should be taken when an interrupt occurs within an interrupt service routine?

   As there are different number and types of devices connected to a computer, processor may receive interrupts from several devices and it is likely that the processor may receive new interrupts while serving an interrupt i.e. while executing the interrupt handler program of an I/O device.
   This issue of having multiple concurrent interrupts pending to be served by processor, can be resolved in two ways:

   1. **Disabled interrupts:** Disable all interrupts while an interrupt is being serviced by processor. The result is that all interrupts received during this period will be set to pending state until processor completes execution of interrupt handler. Upon completion, the next interrupt received in sequence will be serviced. This method is simple to implement as all interrupts will be services in strict sequential order. A drawback of this approach is that it does not consider the relative priorities of the external devices which generate the interrupts and the time-critical needs of various events which generate the interrupts.
   2. **Nested interrupts:** Using this approach, if processor receives interrupt from a device with higher priority than the one which the processor is currently serving, then current state of Interrupt Service Routine (ISR) for low priority device will be saved on stack and processor executes the ISR of higher priority device. Upon completion, processor retrieves the saved state from stack and resumes execution. Otherwise, if a lower priority interrupt is received, then it will be kept pending until completion of current ISR execution for higher priority device. This process is repeated for any number of interrupts received by processor. A benefit of this approach is that relative priorities of interrupts is considered for timely execution of the ISRs of respective devices while the drawback is that the complexity of system design is increased.

2.

   a. Discuss the merits and demerits of using a single bus.

   Merits
   - A single bus is easier to implement and maintain
   - It has a lower initial cost.

   Demerits
   - More the devices attached to the bus, greater the bus length and thereby greater the propagation delay.
   - Bus may become a bottleneck as the aggregate data transfer demand from all connected devices approaches the capacity of bus.

   b. How will the use of a mezzanine bus architecture solve this problem?

   The benefits of this approach are as listed below:
   - A local bus can be used to connect processor and cache, while a system bus can be used to connect cache and main memory; Using this bus hierarchy, I/O transfers to and from the main memory across system bus do not interfere with processor's activity.
   - An expansion bus interface can be used to connect several low speed external I/O devices to system bus. Expansion bus interface buffers data transfers between system bus and the I/O controllers on expansion bus. This allows connection of a wide variety of I/O devices to expansion bus while insulating memory-to-processor traffic from I/O traffic.

- A high speed bus can be used to support high capacity I/O devices while lower sped devices are connected to expansion bus with an interface buffering traffic between expansion bus and high speed bus.
- High speed bus brings high demand devices into closer integration with the processor and at the same time is independent of the processor. Thus, differences in processor and high speed bus speeds and signal line definitions are tolerated. In addition, changes in processor architecture do not affect the high speed bus and vice-versa.

c. Discuss various methods of bus arbitration.

As bus is a shared communication medium to which several components are connected and only one component can successfully transmit data at a time on the bus, **arbitration** needs to be performed. Two approaches of bus arbitration are possible as listed below:

Centralized arbitration:

In the centralized approach, a single hardware device, referred as bus controller / arbiter, is responsible for allocating time on the bus. This device may be a separate module or part of the processor itself.

Distributed arbitration:

In the distributed approach, each of the modules connected to bus contains access control logic and the modules act together to share the bus. An example of this approach is **priority-based distributed arbitration**. In this method, each of the modules is assigned a unique priority value and modules are connected over bus in decreasing priorities with first device on bus having highest priority and last device having lowest priority. A high signal is propagated on the bus and the highest priority node which is interested in data transfer lowers the signal transmitted on bus. This indicates that a higher priority device is interested in data transfer and blocks all lower priority devices from taking control of bus. Thus all devices connected to bus agree on a specific device being master of bus and allowed to transfer data until next bus arbitration cycle.

With either of the arbitration approaches mentioned above, the goal is to designate a device – either processor or an I/O module – as master of bus. The master may then initiate a data transfer on bus to some other module connected to the bus.

3. Consider a hypothetical microprocessor having 64-bit instructions composed of two fields: the first 16-bits contains the op-code and the remainder the immediate operand or an operand address. Assume memory is organized in 32-bit words, i.e. one r/w access can yield a maximum of 32 bits.

a. What is the maximum directly addressable memory capacity (in bytes)?

Given,

32-bit words ==> word length is 32 bits = 4 Bytes

Instruction size is 64 bits; Opcode size is 16 bits ==> Remainder is 64 – 16 = 48 bits for immediate operand or an operand address.

Operand address is 48 bits ==> Total memory size is 2^{48} bytes = **256 TB** 2 points

b. How many bits are needed for the program counter and the instruction register?

Program Counter is the register which holds the address of next instruction to execute in memory. Here, we have 256 TB memory with 48 bits operand address with 2^{48} addressable bytes. Hence Program counter should be 48-bits so that an address to a memory location can be stored in it. 3 points

Instruction Register is the register which holds the instruction to be executed. As the instruction size is 64 bits, the instruction register should be 64 bits size100. 3 points

c. Discuss the impact on the system speed if the microprocessor has: 3 points

(i) a 64-bit local address bus and a 16-bit local data bus, or

Given,
64 bit local address bus
16 bit local data bus
➔ With a 64 -bit address bus, entire 48 bits address of operand can be sent to memory in 1 clock cycle.
➔ With a 16-bit data bus, 4 clock cycles are required to transfer two memory words of 64-bits – which may contain instruction / data.

(ii) a 32-bit local address bus and a 64-bit local data bus. 3 points

Given
32 bit local address bus
64 bit local data bus
➔ With a 32 -bit address bus, 2 clock cycles are required to transfer the 48 bits address of operand. In addition, additional logic/circuitry needs to be built

- into the processor to split the address and issue two separate requests and
- into the memory subsystem to concatenate the two parts to form the 48-bit address and then decode

➔ With a 64 -bit data bus, 1 clock cycle is required to transfer two memory words of 64 – which may contain instruction / data.

4. Consider a hypothetical microprocessor generating a 32-bit address (assume that the program counter and the address registers are 32-bit wide) and having a 32-bit data bus. Assume memory is byte addressable.

a. What is the maximum memory address space that the processor can access directly if connected to a “32-bit memory”?

Given,
32-bit address bus
32-bit data bus
➔ Total memory size is 2^{32} words. 3.5 points
Assuming byte addressable memory, memory directly addressable by processor is 2^{32} Bytes = 4 GB.

b. What is the maximum memory address space that the processor can access directly if connected to an “16-bit memory”?

Given,
32-bit address bus
32-bit data bus
➔ Total memory size is 2^{32} words. 3.5 points
Assuming byte addressable memory, memory directly addressable by processor is 2^{32} Bytes = 4 GB.

c. What architectural features will allow this microprocessor to access a separate “I/O space”?

Similar to the memory access (read/write) instructions issued by processor to access the data/instructions stored at various memory locations, additional instructions need to be included in the instruction set, which include the I/O device address information to be accessed to perform read/write operation. Towards this, additional signal lines may need to be implemented, as appropriate, to identify the I/O devices. 3.5 points

d. If an input and an output instruction can specify a 16-bit I/O port number, how many 16-bit I/O ports can the microprocessor support? How many 32-bit I/O ports?'

3.5 points

With an 16-bit I/O port number, 2^{16} = 65536 I/O ports can be addressed each for input and output operations. i.e. 65536 input devices and 65536 output ports can be addressed.

Depending on the size of the I/O port, the amount of data transferred will vary, i.e. 16-bit I/O ports will transfer 16 bits at a time while 32-bit I/O ports will transfer 32 bits at a time.

Hence the number of ports supported by microprocessor are same in both scenarios.

5. Consider a microprocessor, with a 64-bit external data bus, driven by a 4 *GHz* input clock. Assume this microprocessor has a bus cycle whose minimum duration equals five (5) input clock cycles. What is the maximum data transfer rate that this microprocessor can sustain? To increase its performance, would it be better to make its external data bus *128-bits* or to double the external clock frequency supplied to the microprocessor? Discuss and state any other assumptions you make, and explain. Hint: Determine the number of bytes that can be transferred per bus cycle.

Given,

Input clock rate - 4GHz

==> Clock cycle duration = 1 / 4 GHz = 0.25×10^{-9} sec

Minimum bus cycle duration is 5 input clock cycles = 1.25×10^{-9} sec

Given,

64-bit external data bus

i.e.,

During one bus cycle, 64 bits will be transferred over bus.

Hence the maximum bus data rate

= # of bits transferred per bus cycle / bus cycle duration

= 64 bits / (1.25 x 10^{-9} sec) = 6.4 GB/sec

6 points

Increasing the external data bus size from 64-bits to 128-bits would allow transfer of more number of bits per second, thereby increasing the overall throughput. But, this enhancement should be supported by increased word length of memory to 128-bits so that the read/write operation can be performed in 128-bit words.

4 points

Doubling the external clock frequency of microprocessor may double the overall performance of system provided the number of clock cycles required for execution of instructions does not change, in which case, the actual time taken to execute instructions is reduced by half and hence the performance doubles. This enhancement should be supported by an equivalent increase in performance (speed) of memory module so that the processor does not need to wait for memory operations to complete.

4 points

6. Discuss with a diagram the operation of a synchronous read/write from/to memory. Look up the web for synchronous r/w operation.

A computer system has several modules such as processor, memory, I/O etc. To execute instructions from a program, one or more modules will perform the required operations. These modules coordinate by timing the events generated on bus. Two methods are possible for timing: Synchronous timing and Asynchronous timing.

With synchronous timing, the occurrence of events on the bus is determined by a clock. The bus includes a clock line upon which a clock transmits a regular sequence of alternating 1s and 0s of equal duration. A single 1-0 transmission is referred to as a clock cycle or bus cycle and defines a time slot. all other devices on the bus can read the clock line, and all events start at the beginning of a clock cycle.

Listed below in brief are the tasks performed during a synchronous read/write operation from/to memory:

- Processor places a memory address on address lines during first clock cycle and may assert various status lines.
- Once the address lines have stabilized, processor issues an address enable signal.
- For a read operation, processor issues a read command at the start of second cycle.
- A memory module recognizes the address and after a delay of one cycle, places data on data lines.
- Processor reads data from data lines and drops the read signal.
- For a write operation, processor puts data on data lines at the start of second cycle and issues a write command after data lines have stabilized.
- Memory module copies information from the data lines during third clock cycle.

Shown below is the timing diagram for above memory read/write operation.

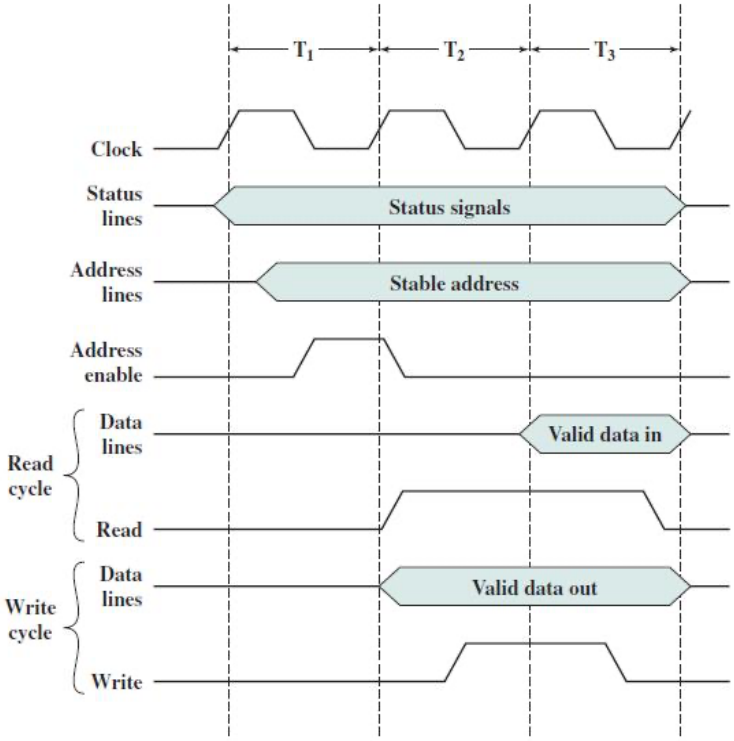

Figure 3.18 Timing of Synchronous Bus Operations

7. Problem 3.8 on Page 117 of the 10th edition of the book.
   Figure 3.26 indicates a distributed arbitration scheme that can be used with an obsolete bus scheme known as Multibus I. Agents are daisy cha-ined physically in priority order. The left-most agent in the diagram receives a constant *bus priority in* (BPRN) signal indicating that no higher-priority agent desires the bus. If the agent does not re- quire the bus, it asserts its *bus priority out* (BPRO) line. At the beginning of a clock cyce, any agent can request control of the bus by lowering its BPRO line. This lowers the BPRN line of the next agent in the chain, which is in turn required to lower its BPRO line. Thus, the signal is propagated the length of the chain. At the end of this chain reaction, there should be only one agent whose BPRN is asserted and whose BPRO is not. This agent has priority. If, at the beginning of a bus cycle, the bus is not busy (BUSY inactive), the agent that has priority may seize control of the bus by asserting the BUSY line.

   It takes a certain amount of time for the BPR signal to propagate from the highest-priority agent to the lowest. Must this time be less than the clock cycle? Explain.

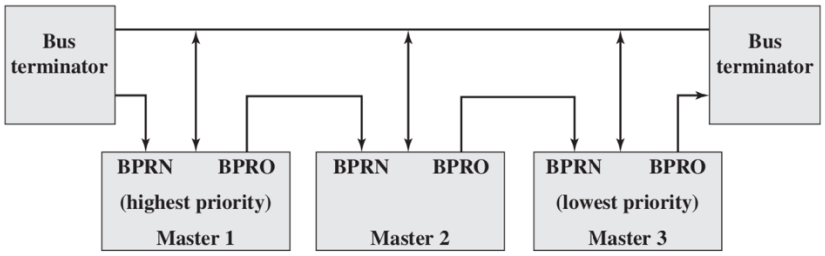

Figure 3.26 Multibus I Distributed Arbitration

Time taken for the BPR signal to propagate from the highest-priority agent to the lowest:

- Initially, the given protocol with respect to daisy chain arrangement says that at the beginning of the clock cycle, there are chances for two or more masters taking the control.
- Here the master with lowest priority will not come to know that master with highest priority has requested the control.
- That is, this leads two or more master to attempt the control on bus simultaneously, if signal is not propagated properly.
- To overcome this disadvantage of accessing control by two or more masters simultaneously the propagation should take less than one clock cycle.

Thus, amount of time taken for BPR signal to propagate from highest priority agent to lowest priority agent must be less than the clock cycle