Problems — 1st batch

1. A CPU is driven by 2 GHz clock.

   (a) Compute the duration of one clock cycle.

   (b) Assume that on average the execution of an instruction takes 4 clock cycles. Compute the performance of the CPU in terms of MIPS (millions of instructions per second).

   (c) Assume that executing a specific program of 400 million instructions takes 2 seconds. How many clock cycles does it take on average to execute an instruction of this program?

   Solution:

   (a) There are $2 \cdot 10^9$ *cycles per second, thus one cycle lasts* $1/(2 \cdot 10^9) = 0.5 \cdot 10^{-9}$ *second, i.e., half nanosecond.*

   (b) The number of instructions executed per second is $2 \cdot 10^9/4 = 500 \cdot 10^6 = 500$ *million, i.e.,* 500 *MIPS.*

   (c) Executing $400 \cdot 10^6$ *instructions takes* $4 \cdot 10^9$ *cycles. Thus* $(4 \cdot 10^9)/(400 \cdot 10^6) = (4/400) \cdot 10^3 = 10$ *cycles are needed for one instruction.*

2. Consider a machine having 32-bit instructions composed of two fields. The first byte contains the opcode and the remainder an immediate operand or an operand address.

   (a) What is the maximum number of operations available?

   (b) What is the maximum directly addressable memory capacity (assuming that every address specifies a byte-long word)?

   (c) Discuss the impact on the system speed if the bus has

      i. a 24-bit address bus and a 32-bit data bus, and

      ii. a 16-bit address bus and a 16-bit data bus.

   (d) How many bits are needed for the program counter and the instruction register?

   Solution:

(a) *The opcode is* 8 *bits long, so there can be* $2^8 = 256$ *different opcodes (operations).*

(b) *The address field is* 24 *bits long, so there are* $2^{24} = 2^4 \cdot 2^{20} = 16 \cdot 2^{20}$ *different addresses. Since every address represents one byte in memory, we get* 16 *MB.*

(c) *In the first case we need one cycle for instruction fetch and one cycle for address transfer. In the second case, two cycles are needed to transfer a* 24*-bit address via an address bus of width* 16 *bits, and similarly two cycles are needed to transfer a* 32*-bit instruction via a data bus of width* 16 *bits.*

(d) *We need a 24 bit (the length of the addresses) PC and a 32 bit (the length of the instructions) IR.*

3. Consider a microprocessor, with a 16-bit data bus, driven by an 8 MHz input clock. Assume that this microprocessor has a bus cycle whose duration equals four input clock cycles. What is the maximum data transfer rate across the bus that the microprocessor can sustain in bits/second (bps)? *Hint:* Determine the number of bits that can be transferred per bus cycle.

   Solution: There are $8 \cdot 10^6/4 = 2 \cdot 10^6$ *bus cycles per second, each time transferring* 16 *bits, thus the maximum transfer rate is* 32 *Mbps (megabits per second).*

4. Consider a 3.2 GHz CPU where executing data processing (arithmetic and logical) instructions takes 4 clock cycles and executing data transfer (load and store) instructions takes 10 clock cycles. When a specific program of one million instructions runs, 60% of the instructions are data processing and 40% of the instructions are data transfer. How long does it take to run this program to completion?

   Solution: It takes $4 \cdot 0.6 \cdot 10^6 + 10 \cdot 0.4 \cdot 10^6 = 6.4 \cdot 10^6$ *cycles, i.e.,* $(6.4 \cdot 10^6)/(3.2 \cdot 10^9) = 2 \cdot 10^{-3}$ *seconds, i.e.,* 2 *milliseconds.*

5. When an arithmetic instruction is executed the execution time is

   - 4 clock cycles if the operands can be fetched from the cache (cache hit)
   - 14 clock cycles if the operands have to be fetched from main memory (cache miss).

The cache hit ratio is 0.6, i.e., 60% of the time the required operands are in the cache. How many clock cycles are needed on average to execute the instruction?

Solution: Consider 100 *instructions. For* 60 *instructions we have cache hits, so we need* $60 \cdot 4 = 240$ *cycles. For* 40 *instructions we have cache misses, so we need* $40 \cdot 14 = 560$ *cycles. Altogether we have* 800 *cycles. Taking the average (dividing by* 100*) we get* 8 *cycles.*

6. A given program needs to read 1 MB data from the hard disk. Data transfer between main memory and the hard disk is done in 1 KB blocks, and it takes 15 ms to read in a block. Determine how much CPU-time is needed for the I/O in case of

   - programmed I/O
   - interrupt-driven I/O
   - using DMA.

   Assume that all the interrupt service procedures needed take 75 ns and you can ignore the initial setup. Would it make sense to schedule the given program while the system is waiting for the read to finish?

   Solution: With programmed I/O the CPU is constantly engaged with the data transfer. Since there are 10^3 *blocks, we get* $10^3 \cdot 15\ ms = 15\ s$. *In the interrupt-driven case, the ISP runs one thousand times (at the end of the transfer of each* 1 *KB block), and the CPU otherwise is free to execute other programs:* $10^3 \cdot 75\ ns = 75\ \mu s$. *With DMA there is only one interrupt (at the end of the whole data transfer):* 75 *ns.*

   The process is probably waiting for input while the read takes place, so it is unlikely to be able to run. It is better to schedule another process during the read.