a) 64 bit instruction ==> 4 bytes or 4 \* 8 = 32 bits opcode , 32 bit address

32 bit instruction ==> 2 bytes or 2 \* 8 = 16 bits opcode , 16 bit address

32 bit instruction ==> 2 bytes or 16 bits opcode , 16 bit address

2^32 = 4294967296 bytes or 4GB memory can be directly addressed

b) Program counter needs 32 bit (32 bits addresses) and instruction register IR needs 64 bits (64 bits addresses) and 64 bits data bus honi chaheye or 64 bit address bus honi cheye

If the address bus is 64 bits, the whole address can be transferred to memory at once and decoded there

**64 bit data bus**

**32 bit data bus**

64-bit local address bus and a 32-bit local data bus. Instruction and data transfers would take three bus cycles each, one for the address and two for the data. Since If the address bus is 64 bits, the whole address can be transferred to memory at once and decoded there; however, since the data bus is only 32 bits, it will require 2 bus cycles (accesses to memory) to fetch the 64-bit instruction or operand.

**16 bit data bus**

64-bit local address bus and a 16-bit local data bus. Instruction and data transfers would take five bus cycles each, one for the address and four for the data. Therefore, the data bus is 16 bits, the microprocessor will need 4 bus cycles to fetch the 64-bit instruction or operand.

c) 32 bits instruction register needs if IR is only contain the opcode and 64 bits instruction register needs to contain the whole instruction.

32-bit local address bus and a 16-bit local data bus. Instruction and data transfers would take three bus cycles each, one for the address and two for the data. Since If the address bus is 32 bits, the whole address can be transferred to memory at once and decoded there; however, since the data bus is only 16 bits, it will require 2 bus cycles (accesses to memory) to fetch the 32-bit instruction or operand.

16-bit local address bus and a 16-bit local data bus. Instruction and data transfers would take four bus cycles each, two for the address and two for the data. Therefore, that will have the processor perform two transmissions in order to send to memory the whole 32-bit address; this will require more complex memory interface control to latch the two halves of the address before it performs an access to it. In addition to this two-step address issue, since the data bus is also 16 bits, the microprocessor will need 2 bus cycles to fetch the 32-bit instruction or operand.

1.5) Since the microprocessor has a 16-bit external data-bus, this means that 16 bits are transferred in 1 bus cycle.

now input clock= 8MHz

This means 8x(10^6) cycles in 1 second. 8000000cycles

therefore, 1 cycle takes 1/(8x(10^6)) seconds 1 / 8000000 = 0.000000125 = 125x10^9 = 125 ns

since 1 bus cycle= 4 clock cycles= 4/(8x(10^6)) seconds 4/8000000 = 0.0000005 =500x10^9 = 500ns

now 1 bus cycle -------> 16 bits or (2 bytes)

4/(8x(10^6)) seconds ---------> (16x8x(10^6))/4 bits = (16x8x(10^6))/(4x8) BYTES

therefore answer= 4x(10^6) Bytes/sec = 4,000,000 = 4Mbytes/sec

Text

Description automatically generated

or

Clock cycle = 1 /8MHz = 125 ns

Bus cycle = 4 × 125 ns = 500 ns

2 bytes transferred every 500 ns; thus transfer rate = 4 MBytes/sec

Or

Text, shape

Description automatically generated with medium confidence

Doubling the frequency may mean adopting a new chip manufacturing technology

(assuming each instructions will have the same number of clock cycles); doubling

the external data bus means wider (maybe newer) on-chip data bus drivers/latches

and modifications to the bus control logic. In the first case, the speed of the memory

chips will also need to double (roughly) not to slow down the microprocessor; in

the second case, the "wordlength" of the memory will have to double to be able to

send/receive 32-bit quantities.

Or

To increase its performance:

By doubling the frequency, it may mean adopting a new chip manufacturing technology (assuming each instruction will have the same number of clock cycles);

By doubling the external data bus, that means wider (maybe newer) on-chip data bus drivers/latches and modifications to the bus control logic.

Therefore, in the first situation thespeed of the memory chips will need to double, not to slow down the microprocessor. Regarding the second situation, the word length of the memory will must double to be able to send/receive 32-bit quantities

1.4) a) TheMaximum memory address space = 2^16 = 64Kbytes

b) The Maximum memory address space= 2^16 = 64Kbytes

Therefore, in (a) and (b), the microprocessor is to access 64K bytes, but the difference thing between them is that the access of 8-bit memory will transfer a 8 bits and the access of 16-bit memory may transfer 8 bits or 16 bits word.

C) Separate I/Oinstructions are needed because during its execution will generate separate its own signals I/O signals. That signals will be different from the memory signalswhich isgenerated duringtheexecutionformemoryinstructions. Therefore, one moreoutput pinwill be needed to carry I/O signals

Or

separate input and output instructions are needed, whose execution will generate separate "I/O signals" (different from the "memory signals" generated with the execution of memory-type instructions); at a minimum, one additional output pin will be required to carry this new signal.

d) With an 8-bit I/O port number the microprocessor can support2^8 = 2568-bit input ports,and 2^8 = 256 8-bit output ports.With an 8-bit I/O port number the microprocessor can support 2^8 = 256 16-bit input ports,and 2^8 = 256 16-bit output ports.Thus, thesize of the I/O port willnot changethe number of I/O ports since the number ofI/O ports depends on the number of bits which is used to representthe I/O port number(equals to8 bitsin both cases).

Or

it can support 28 = 256 input and 28 = 256 output byte ports and the same number of input and output 16-bit ports; in either case, the distinction between an input and an output port is defined by the different signal that the executed input or output instruction generated

1.6) a. Input from the Teletype is stored in INPR. The INPR will only accept data from

the Teletype when FGI=0. When data arrives, it is stored in INPR, and FGI is

set to 1. The CPU periodically checks FGI. If FGI =1, the CPU transfers the

contents of INPR to the AC and sets FGI to 0.

When the CPU has data to send to the Teletype, it checks FGO. If FGO = 0,

the CPU must wait. If FGO = 1, the CPU transfers the contents of the AC to

OUTR and sets FGO to 0. The Teletype sets FGI to 1 after the word is printed.

b) The process described in (a) is very wasteful. The CPU, which is much faster

than the Teletype, must repeatedly check FGI and FGO. If interrupts are used,

the Teletype can issue an interrupt to the CPU whenever it is ready to accept or

send data. The IEN register can be set by the CPU (under programmer control)