Computer architecture, solved problems

v.01 2018/19

1. Minicomputers in the eighties (eg. DEC PDP-11) had 18 address signals and of course, the 18-bit address bus. Answer the following questions:

   a) What was the address space of these computers?
   b) What may have been the largest possible memory of these computers in bytes if the memory location is 1 Byte?
   c) How long should the program counter (PC) of these computers be?
   d) What would be needed to change in these computers if we would like to increase address space 8-times?

   Solution:

   a) 2 ^ 18 = 2 ^ 8 * 2 ^ 10 = 256k (2 ^ 10 = 1K, 2 ^ 20 = 1M, 2 ^ 30 = 1G, 2 ^ 40 = 1T ...)
   b) 2 ^ 18B = 256KB
   c) 18b, because the program can be anywhere in memory ...
   d) add 3 address signals, extend the PC for 3 bits, change the instruction format (the address field extended for 3 bits)

2. Write down a signed decimal number with a value of -25 in 8-bit fixed point presentation in each of the four 8-bit modes for the presentation of signed numbers. Do the same also with the number 33. Write resulting presentations as binary and hexadecimal numbers. Consider, how to perform addition of these numbers in binary form for each presented mode.

   a) sign and magnitude:

   $V(b) = (-1)^{b_{n-1}} \sum_{i=0}^{n-2} b_i 2^i$ + .. 0 / - .. 1 (-127 .. 127)

   $25_{(10)}$ = ?$_{(2)}$ = $11001_{(2)}$
   25 : 2 = 12 + 1
   12 : 2 = 6 + 0
   6 : 2 = 3 + 0
   3 : 2 = 1 + 1
   1: 2 = 0 + 1

   $33_{(10)}$ = ?$_{(2)}$ = $100001_{(2)}$
   33 : 2 = 16 + 1
   16 : 2 = 8 + 0
   8 : 2 = 4 + 0
   4 : 2 = 2 + 0
   2 : 2 = 1 + 0
   1 : 2 = 0 + 1

   therefore: $-25_{(10)}$ = $10011001_{(2)}$ ($99_{(16)}$) $33_{(10)}$ = $00100001_{(2)}$ ($21_{(16)}$)

   addition: it is necessary to take into account the sign => additional complexity...

b) the presentation with offset

$V(b) = \sum_{i=0}^{n-1} b_i 2^i - 2^{n-1}$ tudi $(2^{n-1} - 1)$ -128 .. 127 (-127..128)

$-25+2^{n-1} = -25 + 128 = 103$

$103_{(10)} = ?_{(2)} = 1100111_{(2)}$
103 : 2 = 51 + 1
51 : 2 = 25 + 1
25 : 2 = 12 + 1
12 : 2 = 6 + 0
6 : 2 = 3 + 0
3 : 2 = 1 + 1
1 : 2 = 0 + 1

$3+2^{n-1} = 33 + 128 = 161$

therefore: $-25_{(10)} = 01100111_{(2)}$ $(67_{(16)})$ $33_{(10)} = 10100001_{(2)}$ $(A1_{(16)})$

addition: the result includes two offsets, so one offset has to be subtracted separately

c) ones' complement

$V(b) = \sum_{i=0}^{n-1} b_i 2^i - b_{n-1}(2^n - 1)$ -127 .. 127

$25_{(10)} = 11001_{(2)}$

therefore: $-25_{(10)} = 11100110_{(2)}$ $(E6_{(16)})$, $33_{(10)} = 00100001_{(2)}$ $(21_{(16)})$

addition: in case of carry from place *n-1* , 1 has to be added to result; on the other hand, sign bit can be treated the same as the other (value) bits ...

d) two's complement

$V(b) = \sum_{i=0}^{n-1} b_i 2^i - b_{n-1}(2^n)$ -128 .. 127

$25_{(10)} = 00011001_{(2)}$

$11100110_{(2)}$
$+00000001_{(2)}$
$=11100111_{(2)}$

therefore: $-25_{(10)} = 11100111_{(2)}$ $(E7_{(16)})$ $33_{(10)} = 00100001_{(2)}$

3. Write the value of 4.75 in 32-bit floating point format IEEE 754. Another floating point value is written in the memory as $44FAC000_{(16)}$. Which value does it represent if it is written in the format IEEE 754? (This problem is informative, similar floating point problems will not appear in written exams)

$4_{(10)} = ?_{(2)} = 100_{(2)}$
4 : 2 = 2 + 0
2 : 2 = 1 + 0
1 : 2 = 0 + 1

$0{,}75_{(10)} = ?_{(2)} = 0{,}11_{(2)}$
0,75 * 2 = 0,5 + 1
0,5 * 2 = 0 + 1

$4{,}75_{(10)} = 100{,}11_{(2)}$

In floating point: 4,75 = -1^s*m*2^exp = 1 * 100,11 * 2^0
mantissa is converted into a form of 1, ???:
100,11 * 2^0 = 1,0011 * 2^2

s = 0
m = 1,0011
exp = 2

IEEE 754:

s	exp+127	m
1	8	23

s=0

exp written in a presentation with offset 127:
$exp + 127 = 129_{(10)} = 10000001_{(2)}$

mantissa it always in a form of 1, ???, the implicit bit (1,) is not part of the presentation:
m = 0011

the final presentation result:
$0\underline{10000001}00110000000000000000000 = 40980000_{(16)}$

Second problem: The conversion in the opposite direction:

$44FAC000_{(16)} = 0\underline{10001001}11110101100000000000000_{(2)}$

s = 0
$exp+127=10001001_{(2)} = 137$,
therefore exp=10 , m=1,111101011 $_{(2)}$

2^0*m*2^exp
1*1,111101011*2^10
$11111010110_{(2)}$ = **2006 $_{(10)}$**

4. We want to compare the computers R1 and R2, which differ that R1 has the machine instructions for the floating-point operations, while R2 has not (FP operations are implemented in the software using several non-FP instructions). Both computers have a clock frequency of 400 MHz. In both we perform the same program, which has the following mixture of commands:

Type the command	Dynamic Share of instructions in program (p_i)	Instruction duration (Number of clock periods CPI_i)	
		R1	R2
FP addition	16%	6	20
FP multiplication	10%	8	32
FP division	8%	10	66
Non - FP instructions	66%	3	3

a) Calculate the MIPS for the computers R1 and R2.

b) Calculate the CPU program execution time on the computers R1 and R2, if there are 12000 instructions in the program?

c) At what mixture of instructions in the program will both computers R1 and R2 be equally fast?

Solution:

a) CPI = $\sum_{i=0}^{3} CPI_i * p_i$

MIPS = $\frac{f_{CPE}}{CPI*10^6}$

Computer R1:

$$\text{CPI} = \sum_{i=1}^{3} CPI_i * p_i = 0{,}16 * 6 + 0{,}1 * 8 + 0{,}08 * 10 + 0{,}66 * 3 = 4{,}54$$

Computer R1 needs an average of 4.54 clock periods for one instruction

$$\text{MIPS} = \frac{f_{CPE}}{CPI*10^6} = \frac{400*10^6}{4{,}54*10^6} = 88{,}1$$

Computer R1 executes an average of 88 100 000 instructions per second.

Computer R2:

$$\text{CPI} = \sum_{i=1}^{3} CPI_i * p_i = 0{,}16 * 20 + 0{,}1 * 32 + 0{,}08 * 66 + 0{,}66 * 3 = 13{,}66$$

Computer R2 needs an average of 13.66 clock periods for one instruction

$$\text{MIPS} = \frac{f_{CPE}}{CPI*10^6} = \frac{400*10^6}{13{,}66*10^6} = 29{,}28$$

Computer R2 executes an average of 29 280 000 instructions per second.

b) $$\boldsymbol{CPU}_{\text{time}} = \frac{\text{Number\_of\_instructions}}{\boldsymbol{MIPS} * \mathbf{10^6}}$$

Another form of the equation to calculate the CPU time is:

$$CPU_{time} = Number\_of\_instructions * CPI * t_{CPU}$$

Computer R1:

$$\boldsymbol{CPU}_{\text{time}} = \frac{\text{Number\_of\_instructions}}{\boldsymbol{MIPS} * \mathbf{10^6}} = \frac{12000}{88{,}1 * 10^6} = 136{,}2 * 10^{-6} = 136{,}2\ \mu s$$

Computer R2:

$$\boldsymbol{CPU}_{\text{time}} = \frac{\text{Number\_of\_instructions}}{\boldsymbol{MIPS} * \mathbf{10^6}} = \frac{12000}{29{,}28 * 10^6} = 410 * 10^{-6} = 410\ \mu s$$

c) Programs without FP instructions ...

5. We want to speed up computer performance with an additional unit for calculating in floating point format. This unit is 20 times faster than the same operations without unit. What percentage of a total computer time must this unit be active to achieve an overall increase in computer speed for 2.5 times?

Solution:

Use Amdahl's law:

$$S(N) = \frac{N}{1 + (N-1) * f}$$

f	=	share of operations that do not speed up
N	=	Factor of speedup for (1 - f) share of operations
S (N)	=	Increase of the overall speed

In this case, is S (N) = 2.5; N = 20, however, we are looking for (1 - f)

$$1 - f = \frac{N - S(N)}{S(N) * (N-1)} = 1 - \frac{20 - 2,5}{2,5 * 19} = 1 - 0,3684 = 0,6315$$

FP unit must be active 63.15% of the time so that the computer's performance is 2.5 times faster.

6. The computer has a main memory access time of 60 ns. We want to reduce this time to 20 ns by adding cache. Determine how fast the cache must be (access time) if we can expect a 90% probability of a hit.

t_{ag}=60 ns
t_a=20 ns
H=90%=0,9
t_{ap}=?

$t_a = t_{ap} + (1-H) \times t_{ag}$
$t_{ap} = t_a - (1-H) \times t_{ag}$
$t_{ap} = 20 \times 10^{-9}$ [s] $- (1-0,9) \times 60 \times 10^{-9}$ [s] $= 20 \times 10^{-9} - 6 \times 10^{-9} = 14 \times 10^{-9}$ [s] = 14 [ns]

7. In a computer with cache, we have the average number of clock periods per instruction equal to 4, if there are no misses in the cache.

a) What is the real number of clock periods per instruction, if the probability of miss in the cache is 10%? For the replacement of the block (line) in the cache, we need 5 clock periods for read and 10 for write accesses. Assume that each instruction requires an average of 2 memory accesses and that 20% of all are write accesses.

$CPI_I=4$
$(1-H)=10\%=0,1$
$M_I=2$
$N_R=5$
$N_W=10$
$P_W=0,2$
$P_R=0,8$

$CPI_R=CPI_I+M_I\times(1-H)\times miss\_penalty$

$miss\_penalty = P_W\times num\_periods\_for\_write + P_R\times num\_periods\_for\_read = P_W\times N_W + P_R\times N_R$

$CPI_R= 4 + 2 \times 0.1 \times (0.2 \times 10 + 0.8 \times 5) = 5.2$

b) What is the real CPI, if we increase the probability of hit to 95%?

$CPI_I=4$
$M_I=2$
$(1-H)=0,05$
$miss\_penalty =6$
$CPI_R=?$

$CPI_R=CPI_I+M_I\times(1-H)\times miss\_penalty$
$CPI_R=4+2\times0,05\times6=4,6$

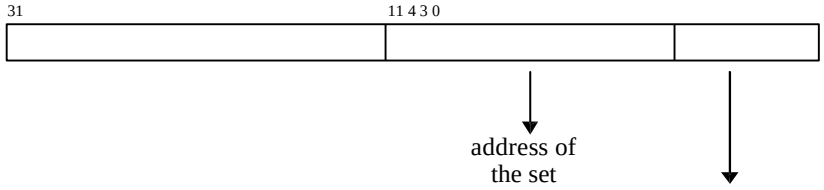
8. On a computer with a 32-bit memory address and the length of the memory location of 1 byte is installed set-associative cache. Cache size is 16 KB, block (line) size is 16 Bytes, set associative cache is 4-way.

a) How many sets are there in cache?

n=32
M=16KB=2^4KB=2^{14}B
block=B=16B=2^4B
E=4=2^2

M=S×E×B
S=M÷(E×B)=2^{14}÷(2^2×2^4)=2^{14-6}=2^8=256 sets

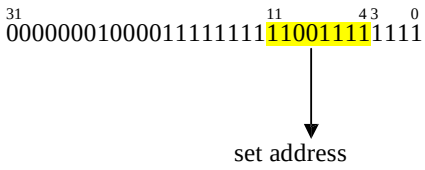
b) Which bits in the memory address determine the address of the set?



Address bits of set: 4-11.

c) Into which set is mapped the content of the memory address 10FFCFF$_{(HEX)}$?

1 0001
2 0010
3 0011
4 0100
5 0101
6 0110
7 0111
8 1000
9 1001
A 1010
B 1011
C 1100
D 1101
E 1110
F 1111



Address belongs to set 207.

9. A computer with virtual memory has an access time tomain memory 50 ns, the time to transfer a block from the virtual into main memory is 10 ms. The probability for the page-fault is 10^{-6}.

What is the average access time, if the page-table is in the main memory?
t_{ag}=50ns
t_B=10ms
(1-H)=10^{-6}
t_a=?

$t_a=t_{ag}+t_{ag}+(1-H)\times t_B$

(access to the page-table in main memory)

(access to the page-frame)

$t_a=50\times10^{-9}+50\times10^{-9}+10^{-6}\times10\times10^{-3}=$
$=100\times10^{-9}+10\times10^{-9}=110\times10^{-9}=110$ [ns]

10. Computer with virtual memory has the following features:
    - length of the virtual address is 38 bits,
    - the page size is 16 KB,
    - length of the physical address is 32 bits.

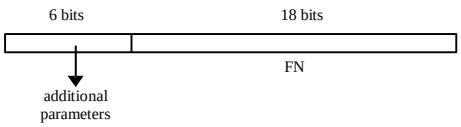
a) How many bits is the length of page descriptor, if in addition to the frame number (FN), additional parameters occupy another 6 bits?

n=38
f=32
page size = 16 KB=$2^p=2^{14}$B

- a number of pages in virtual memory:
  $2^{n-p}=2^n\div2^p=2^{38}\div2^{14}=2^{24}$
- a number of page frames in main memory:
  $2^{f-p}=2^f\div2^p=2^{32}\div2^{14}=2^{18}$ (FN)

Page descriptor = frame number (FN) + 6

Page descriptor = 24 bits = 3 B



b) What is the maximum size of the page-table in bytes?

num_pages x page_descriptor

2^{24}3B x = $2^{20}\times 2^4$3B x = $2^4\times$ 3MB = 16 × 3MB = 48MB page-table size in main memory.