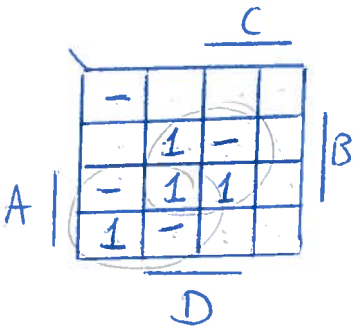
CAO exam 26 Sept. 2014

Q1

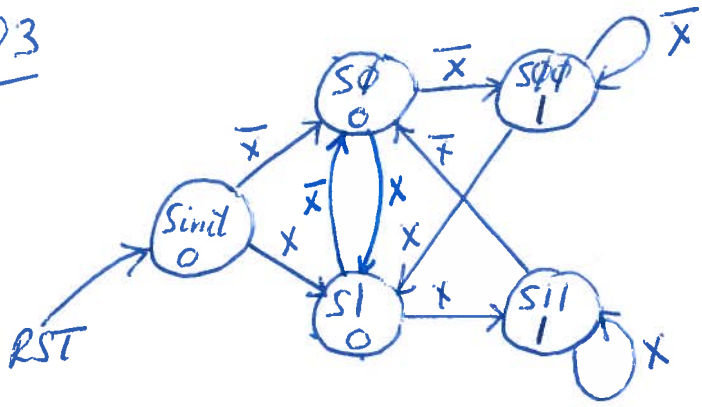


$A \cdot \overline{C} + B \cdot D$

Q2 see documentation ARC processor page 2 (fig 5.4)

Condition code is updated when $\overline{F_3} \cdot \overline{F_2}$

Q3



Q4 point left of hidden bit $0.1\underbrace{xxx\ldots x}_{121\ bits}$

$0.5 \le fraction < 1$

$-2.8 = \left(\frac{-2.8}{4}\right) \cdot 2^2 = \underset{\uparrow}{-}\ \underset{\uparrow}{0.7} \cdot 2^{\overset{(2)}{\uparrow}}$ — exponent

sign, fraction

$0.7 \times 2 = 1.4$	1
$0.4 \times 2 = 0.8$	0
$0.8 \times 2 = 1.6$	1
$0.6 \times 2 = 1.2$	1
$0.2 \times 2 = 0.4$	0
$0.4 \times 2 = 0.8$	0

Repeating

fraction $.1\underbrace{0}_{\text{hidden bit}}1001100\overline{11}\ldots$

exponent $2 + 30 = 32_{10}$ $\quad 00001\,00000_2$

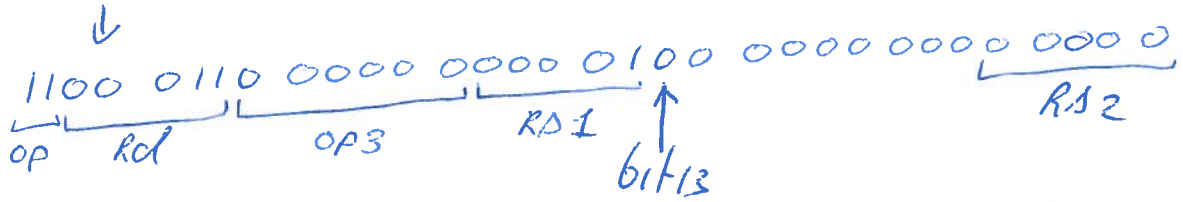
sign = 1

exponent = 0000100000

fraction field = 01100110011... (Repeating)

Q5 Documentation APC page 2 (fig 5-2) (2)

C6004000



assembly instruction is LD [%R1 + %R0], %R3
since %R0 is always 0, also correct

LD [%R1 + 0], %R3 or
LD [%R1], %R3 or
LD %R1, %R3

Q6 Run the program!

address	data
200 - 203	16
204 - 207	-9
208 - 211	13
212 - 215	12
216 - 219	0

(5)

Q7

X	F_1	F_0	F_1^t	F_0^t	Z
0	0	0	0	0	0
0	1	0	i	1	0
0	1	1	0	0	0
1	0	0	1	0	i
1	1	0	1	0	1
1	1	1	1	1	1

DF_1

X \ F_1	0	-	0	1
1	1	-	1	1

(F_0)

$\rightarrow DF_1 = X + F_1 \cdot \overline{F_0}$

$Z = X$

Q8

op = 16
op3 = 000111

} ARC documentation page 3 (fig 5-13)

start addrem

$110\,000111\,00_2 = 1564_{10}$

addrem	A	Amux	B	Bmux	C	Cmux	Rd	Wr	ALU	Cond	Jumpaddrem
1564	-	1	-	1	%tempØ	Ø	Ø	Ø	addcc	next	-
1565	%tempØ	Ø	-	1	-	1	Ø	Ø	addcc	jump	2047

Alternatives are possible, e.g. addl instead of addcc

9a Dynamic RAM.

Consists of capacitors which lose charge. Periodically data is read and written back to recharge the capacitors.

b. - Stored Program Concept.
- Consists of 5 elements which are interconnected: Memory, Control, ALU, Input and output.

c Processor polls the I/O device and to determine whether the I/O device (module) has data available (in case of a read operation) or can accept data (in case of a write operation). So, all I/O is done by the CPU.

10 Printer requires $100\% \cdot 1000\,\mu s \cdot 1/(4000\,\mu s)$
$= 25\%$ of the processing capacity.

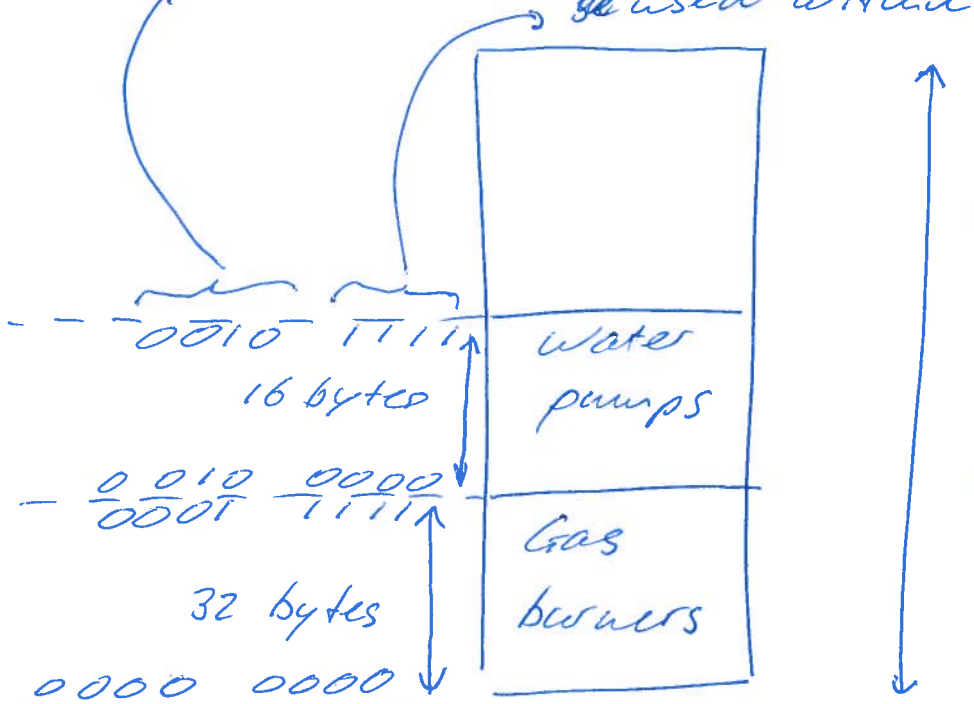
Disk requires $100\% \cdot 125\,\mu s \cdot 1/(1000\,\mu s)$
$= 12.5\%$

Display requires $100\% \cdot 100\,\mu s \cdot 1/(1000\,\mu s)$
$= 10\%$

In total, the processor is occupied with I/O for $25 + 12.5 + 10 = 47.5\%$ of the time which means that 52.5% is left for Program P. Thus fraction of time processor available $= 0{,}525$.

A task of 100 s then takes $100/0.525 \approx 190{,}55$

11 used to select Waterpump I/O space



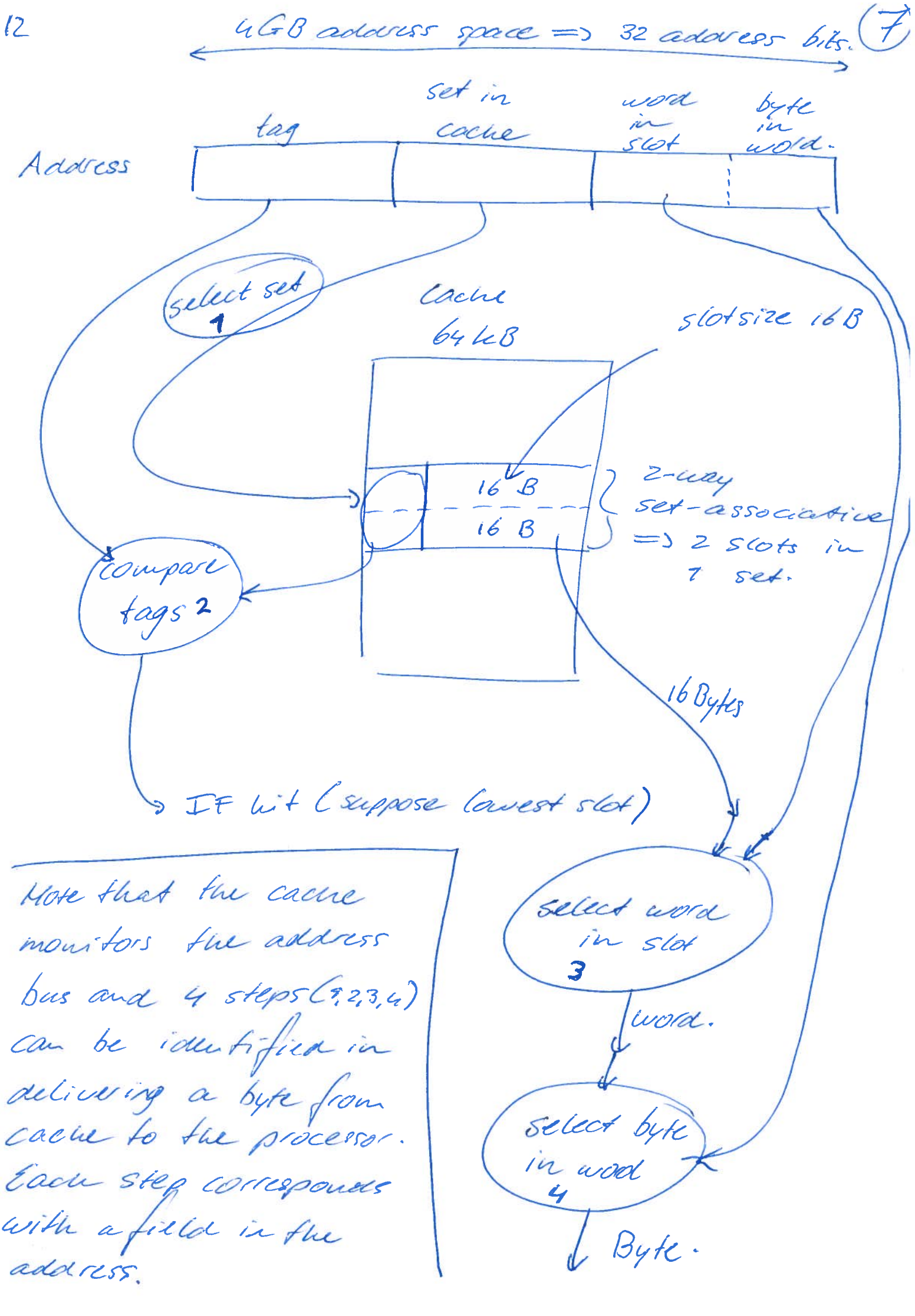
used address within the Gas burner I/O space

used to select the Gas burner I/O space ⇒

$SelBurn = !A_7 . !A_6 . !A_5 . !M/I_O$

$SelWater = !A_7 . !A_6 . A_5 . !A_4 . !M/I_O$

These cannot be further minimized because shadowing is not allowed.


4GB address space => 32 address bits.
7
tag
set in cache
word in slot
byte in word.
Address
select set 1
cache 64kB
slotsize 16B
16 B
16 B
2-way set-associative => 2 slots in 1 set.
compare tags 2
16 Bytes
If hit (suppose lowest slot)
select word in slot 3
word.
select byte in word 4
Byte.
Note that the cache monitors the address bus and 4 steps (1,2,3,4) can be identified in delivering a byte from cache to the processor. Each step corresponds with a field in the address.

12: Calculations

64 kB cache

slotsize 16 B

2 way set associative → 32 B/set → 2 k sets

⇓

11 address bits.

16-bits microprocessor ⇒ 16 bit data bus ⇒ wordsize 16 bit = 2 B.

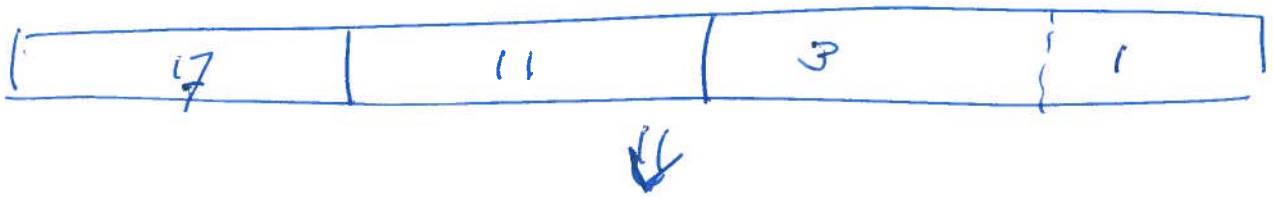
To select 1 word (of 2B) out of a slot with 8 words (= 16 B) takes 3 address bits

---

Selection of 1 B out of a word of 2 B requires 1 address bit

In total 11 + 3 + 1 bits are used ⇒

32 − (11 + 3 + 1) = 32 − 15 = 17 bits remain for the tag



13 (9)

Probability of a Jump instruction = $P_j = 1/5 = 0,2$.

Probability a Jump is taken = $P_t = 0,3$.

Branch penalty = $P = 2$

Average instruction time = $1 + P_j \cdot P_t \cdot P =$

$1 + 0,2\ 0,3 \cdot 2 = 1,12$