Guidelines

- Do **not** write your answers on this sheet. Responses written here will **not** be evaluated. Please use the separate answer sheet provided.
- The attached appendices may assist you with some questions.
- You may use these sheets for rough work, but they must be submitted at the end of the exam.
- Each question is worth 2 points. Total marks: 100 points. If you find any question difficult, proceed to the next one.

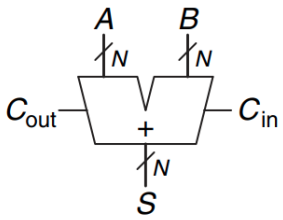
A1

Complete the truth table for OR

```
Unset
    A | B | Output (A OR B)
     -------------------------
     0 | 0 | ?
     0 | 1 | ?
     1 | 0 | ?
     1 | 1 | ?
```

Give as your answer four digits without spaces - the values from row 1st to row 4th of the truth table. For example, `0000`.

A2



You built an 8-bit Adder based on the Full Adder logic. If the inputs values are:

```
Unset
A = 10101010, B = 01010101
```

What will the output **S** value be? Give answer in form of 8 digits without spaces

A3

Using the ALU designed in you homework, what will the output `out` be when ALU inputs are:

```
Unset
x = 11..1111, y = 00..0001, zx = 0, nx = 0, zy = 0, ny = 0, f = 0, no = 0
```

Give as an answer 4 lowest bits of the `out`. For example `0000`

B1

How this instruction will be presented in memory: `@10`

A. `0101010101010100`
B. `0000000000000010`
C. `0100000000000000`
D. `0000000000001010`

B2

And are given the following program:

```
Unset
@10
M=A
D=1
D=D+M
A=D
```

Give as an answer value of `A` register after executing this program. Give one decimal number, for example: `10`

C1

Which instruction takes **less** cycles to be executed in the multicycle MIPS processor?

A. Jump (J)
B. R-type
C. Store (SW)
D. Load (LW)

C2

What is the WAR hazard in a pipelined processor?

A. When a processor's clock speed decreases to wait for slow read/write memory operation
B. When read and write instructions accidentally swap places due to branch prediction errors
C. When the processor runs out of empty registers to write during execution
D. When a subsequent instruction writes to a register before a previous instruction reads from it

---

D1

Suppose access times of 1, 50, and 100 cycles for the L1 cache, L2 cache, and main memory
Assume that the L1 and L2 caches have miss rates of 10% and 20%, respectively
In this case average memory access time in cycles is:

A. 10
B. 42.66
C. 8
D. 4.5

D2

What is the difference between Paging and Segmentation?

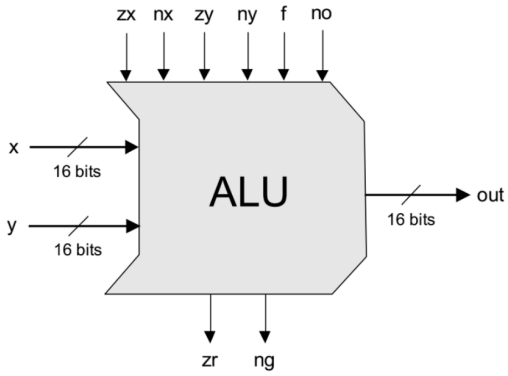
A. Paging allows to use any number of linear address spaces
B. Segmentation is invisible for programmers
C. Segments are never unloaded to disk, they always remain in RAM
D. Size of segments, unlike pages, can be varied

D3

Which of the following is a way to resolve control conflicts in a pipeline processor:

A. Commands transcoding
B. Pipeline reversing
C. Forwarding
D. Reloading commands

Appendix 1. Hack processor ALU



pre-setting the x input		pre-setting the y input		selecting between computing + or &	post-setting the output	Resulting ALU output
zx	**nx**	**zy**	**ny**	**f**	**no**	**out**
if zx then x=0	if nx then x=!x	if zy then y=0	if ny then y=!y	if f then out=x+y else out=x&y	if no then out=!out	out(x,y)=
1	0	1	0	1	0	0
1	1	1	1	1	1	1
1	1	1	0	1	0	-1
0	0	1	1	0	0	x
1	1	0	0	0	0	y
0	0	1	1	0	1	!x
1	1	0	0	0	1	!y
0	0	1	1	1	1	-x
1	1	0	0	1	1	-y
0	1	1	1	1	1	x+1
1	1	0	1	1	1	y+1
0	0	1	1	1	0	x-1
1	1	0	0	1	0	y-1
0	0	0	0	1	0	x+y
0	1	0	0	1	1	x-y
0	0	0	1	1	1	y-x
0	0	0	0	0	0	x&y
0	1	0	1	0	1	x|y

Appendix 2. Hack language commands encoding

A instruction

Symbolic: `@`*xxx* (*xxx* is a decimal value ranging from 0 to 32767, or a symbol bound to such a decimal value)

Binary: `0` *vvvvvvvvvvvvvvv* (*vv ... v* = 15-bit value of *xxx*)

C instruction

Symbolic: *dest* = *comp* `;` *jump* (*comp* is mandatory. If *dest* is empty, the = is omitted; If *jump* is empty, the `;` is omitted)

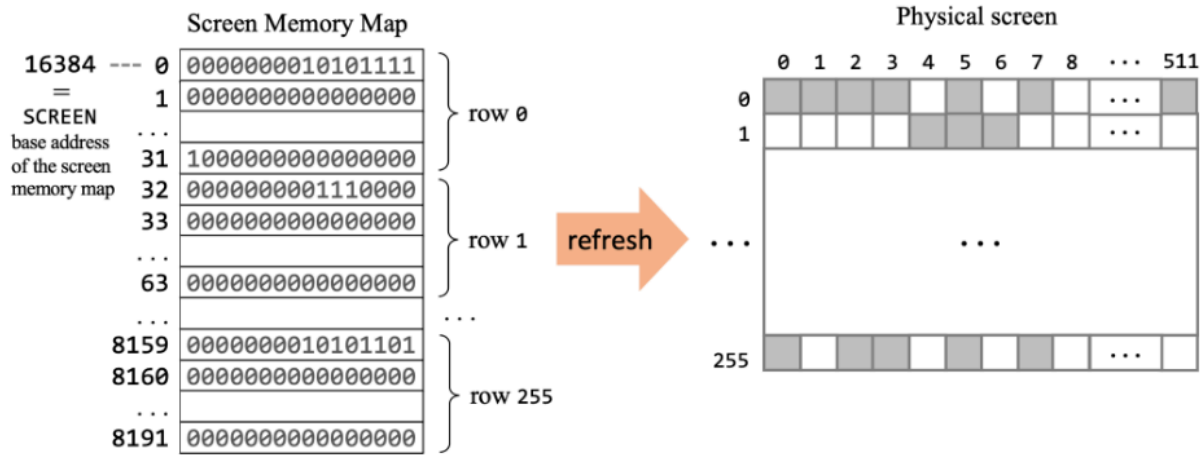
Binary: `111`*accccccdddjjj*

comp		*c*	*c*	*c*	*c*	*c*	*c*
0		1	0	1	0	1	0
1		1	1	1	1	1	1
-1		1	1	1	0	1	0
D		0	0	1	1	0	0
A	M	1	1	0	0	0	0
!D		0	0	1	1	0	1
!A	!M	1	1	0	0	0	1
-D		0	0	1	1	1	1
-A	-M	1	1	0	0	1	1
D+1		0	1	1	1	1	1
A+1	M+1	1	1	0	1	1	1
D-1		0	0	1	1	1	0
A-1	M-1	1	1	0	0	1	0
D+A	D+M	0	0	0	0	1	0
D-A	D-M	0	1	0	0	1	1
A-D	M-D	0	0	0	1	1	1
D&A	D&M	0	0	0	0	0	0
D|A	D|M	0	1	0	1	0	1
a == 0	*a* == 1						

dest	*d*	*d*	*d*	Effect: store *comp* in:
null	0	0	0	the value is not stored
M	0	0	1	RAM[A]
D	0	1	0	D register (reg)
DM	0	1	1	RAM[A] and D reg
A	1	0	0	A reg
AM	1	0	1	A reg and RAM[A]
AD	1	1	0	A reg and D reg
ADM	1	1	1	A reg, D reg, and RAM[A]

jump	*j*	*j*	*j*	Effect:
null	0	0	0	no jump
JGT	0	0	1	if *comp* > 0 jump
JEQ	0	1	0	if *comp* = 0 jump
JGE	0	1	1	if *comp* ≥ 0 jump
JLT	1	0	0	if *comp* < 0 jump
JNE	1	0	1	if *comp* ≠ 0 jump
JLE	1	1	0	if *comp* ≤ 0 jump
JMP	1	1	1	unconditional jump

Appendix 3. Hack computer screen memory mapping



Appendix 4. Complete multicycle MIPS processor