

第一题：Using Nor gate to build and,or, Not gate

这有谷歌上的一个截图， 就是这样的， 链接是：

https://en.wikipedia.org/wiki/NOR_logic， 也许会考用别的门转， 我是用这个公式做的

$$(22) \overline{A+B} = \overline{A} \cdot \overline{B}$$

$$(23) \overline{A \cdot B} = \overline{A} + \overline{B}$$

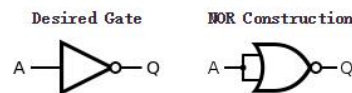
Making other gates by using NOR gates [edit]

A NOR gate is a universal gate, meaning that any other gate can be represented as a combination of NOR gates.

NOT [edit]

This is made by joining the inputs of a NOR gate. As a NOR gate is equivalent to an OR gate leading to NOT gate, this automatically sees to the "OR" part of the NOR gate, eliminating it from consideration and leaving only the NOT part.

output= NOT(A+A)



Truth Table

Input A	Output Q
0	1
1	0

OR [edit]

The OR gate is simply one NOR gate followed by a second whose inputs are joined.

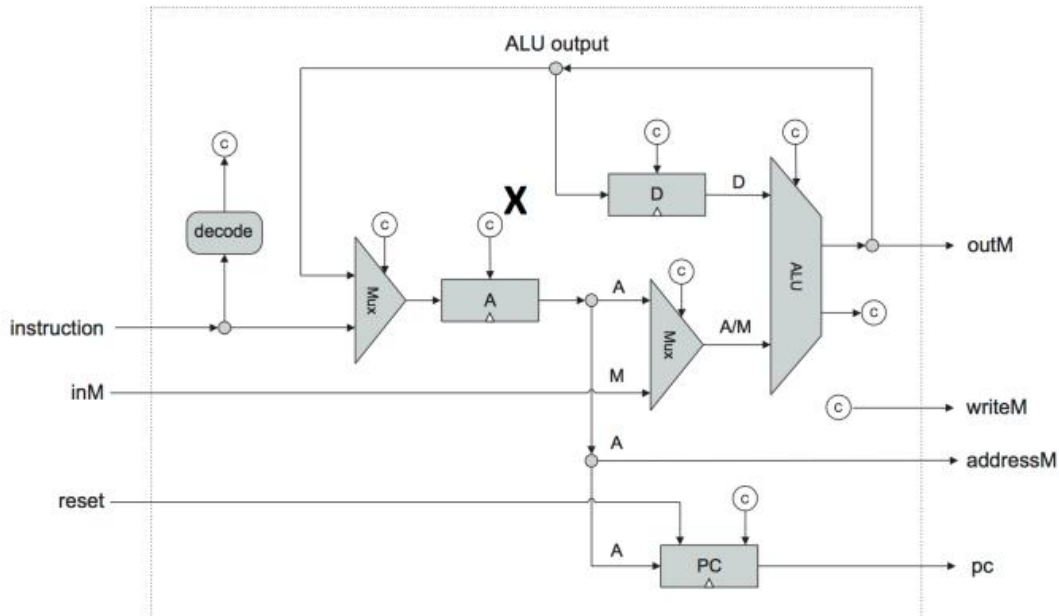
剩下的小问里考了 dmux 和 mux 在 RAM 中的作用，好像是 ram 吧，记不住了，一个分路一个多路复用，实在记不清了

第二题：ALU 那个大表格，这题当时好像都不会做， 后来补考的时候才会，考得好像是如果 $f(x,y)=x-y$ ， zx, nx, zy 这六个指标分别取得是什么值

These bits instruct how to preset the x input		These bits instruct how to preset the y input		This bit selects between + / And	This bit instructs how to postset out	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	f(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

第四题：Hack 语言转 VM，这题具体的回忆不起来了
好像问了@i 这样的是什么意思或者转为二进制

第五题，CPU，问了 MUX 那个 C 和 ALU 右边那个



第六题：补全表格，把 a=1 的时候的那一列补全，好像是左右一样，你需要去确认一下

(when a=0)	c1	c2	c3	c4	c5	c6	(when a=1)
comp mnemonic							comp mnemonic
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	1	1	0	0	0	0	M
!D	0	0	1	1	0	1	
!A	1	1	0	0	0	1	!M
-D	0	0	1	1	1	1	
-A	1	1	0	0	1	1	-M
D+1	0	1	1	1	1	1	
A+1	1	1	0	1	1	1	M+1
D-1	0	0	1	1	1	0	
A-1	1	1	0	0	1	0	M-1
D+A	0	0	0	0	1	0	D+M
D-A	0	1	0	0	1	1	D-M
A-D	0	0	0	1	1	1	M-D
D&A	0	0	0	0	0	0	D&M
D A	0	1	0	1	0	1	D M

第七题：VM 转 HACK： Function bob 4
记住四种情况应该就行了,区别在于参数的数量

Pop local 2

@sp

AM=M-1

D=M

@LCL

A=M+1

push constant 7

@7

D=A

@sp

A=M

M=D

function f 1

@sp

A=M

M=0

A=A+1

M=0

if-goto bob

@sp

AMD=M-1

D=M

@sp

D;JNE

M+D

@sp

M=M+1

第八题：画栈堆的那个表，你看看发生变化时这个表是怎么变化的

.	ARG	Argument 0
		Return addr
		LCL
		ARG
		This
		That

第九题：像是 mock 上那样 $d = (2 + x) * (y - 5)$

这题可特么阴了，考试的时候是除号 $d = (2 + x) / (y - 5)$

括号括在不同的地方会有不同的值，有一个问题就是，当你用一个数除以另一个数的时候要取整，我们考的时候大家都没取整，好像都错了，最终的答案都是整数。

第十题：Jack 语言转换成 HML

第十一题：翻译下面这四行

b=complex.new(1,5)

aa=Array.new(5)

Return

Let bob =complex.add(a,b)

你参考下答案，第一行

第二行

第三行

Push constant 1

push constant 5

push constant 0

Push constant 5

call Array.new 1

return

Call complex.new 2

pop constant 0

Pop argument 1

第四行：

Push argument 0

Push argument 1

Call complex.add 2

Pop..1 这一行答案 pop 和 1 之间那个词不是很清楚，你自己查一下

12. 多线程的有点