EEM16 Midterm

TOTAL POINTS

87.5 / 100

QUESTION 1

Problem 1 30 pts

almost correct...

1.1 (a,b,c) **11.5 / 18**

✓ **- 3 pts** (b) incorrect # prime implicants by 2

✓ **- 2 pts** (b) incorrect size of prime implicant

✓ **- 1.5 pts** (c) incorrect # essential by 1

1.2 (d,e) **12 / 12**

✓ **- 0 pts** Correct

QUESTION 2

2 Problem 2 (a,b) 16 / 20

✓ **- 2 pts** missing a+!d+!e

✓ **- 2 pts** missing a+!b+!c+!d

QUESTION 3

Problem 3 28 pts

3.1 (a,b,c,d) **16 / 16**

✓ **- 0 pts** correct from (a)-(d)

3.2 (e,f,g) **10.5 / 12**

✓ **- 1 pts** e2 - can have both positive and negative exponents easily, negative exponents give more accuracy

✓ **- 0.5 pts** f2 - borrow bit required in msb -> underflow

QUESTION 4

Problem 4 22 pts

4.1 (a) **8 / 8**

✓ **- 0 pts** Correct

4.2 (b) **7 / 7**

✓ **- 0 pts** Correct

4.3 (c) **6.5 / 7**

✓ **- 1.5 pts** incorrect boundary condition

+ 1 Point adjustment

Midterm Exam

Name (Last, First):

Student Id #:

Do not start working until instructed to do so.

1. *You must answer in the* ***space provided*** *for answers after every question. We will ignore answers written anywhere else in the booklet.* ***All pages in this booklet must be accounted*** *for otherwise it will not be graded.*
2. *You are permitted 1 page of notes 8.5x11 (front and back).*
3. *You may not use any electronic device.*

Following table to be filled by course staff only

	Maximum Score	Your Score
Question 1		
Question 2		
Question 3		
Question 4		
TOTAL	100	

Question #1

Consider the Boolean function defined by the truth table below where *A, B, C, and D* are inputs, and *Y* is the sole output.

	A	B	C	D	Y
0	0	0	0	0	0 ✓
1	0	0	0	1	X
2	0	0	1	0	1
3	0	0	1	1	0 ✓
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	0 ✓
7	0	1	1	1	0 ✓
8	1	0	0	0	0 ✓
9	1	0	0	1	X
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	X
14	1	1	1	0	1
15	1	1	1	1	1

(a) Complete the following statements

$$\neg Y = \sum m(0, 3, 6, 7, 8 \qquad)$$

(b) Complete the Karnaugh Map shown below for ~Y, **circle** the prime implicants.

AB

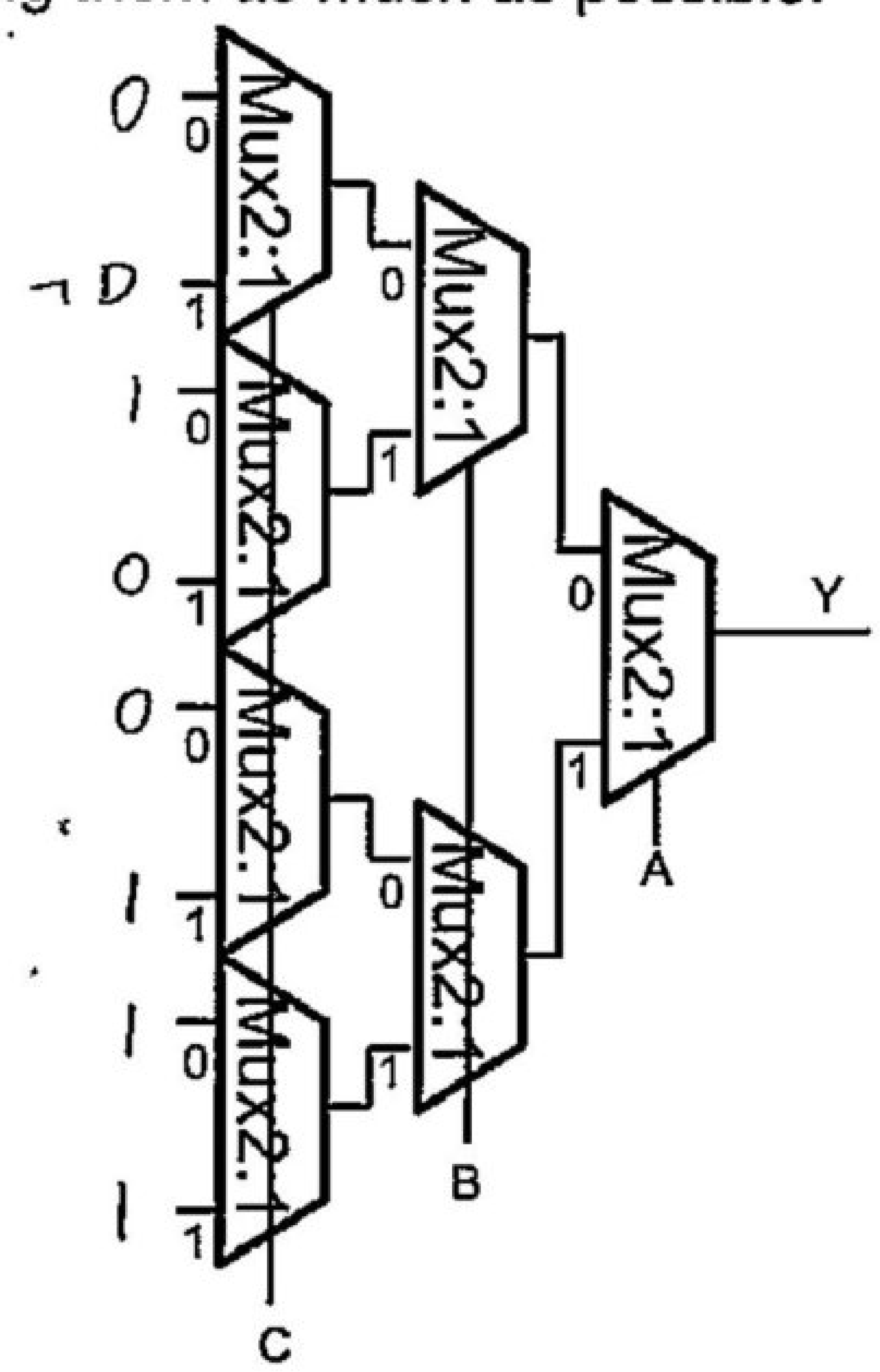
CD	"00"	"01"	"11"	"10"
"00"	1 0	0 4	0 12	1 8
"01"	X 1	0 5	X 13	X 9
"11"	1 3	1 7	0 15	0 11
"10"	0 2	1 6	0 14	0 10

How many prime implicants are there? 7

(c) Write the Boolean (sum-of-product) expression of just the *essential* prime implicants of (b) (if any).

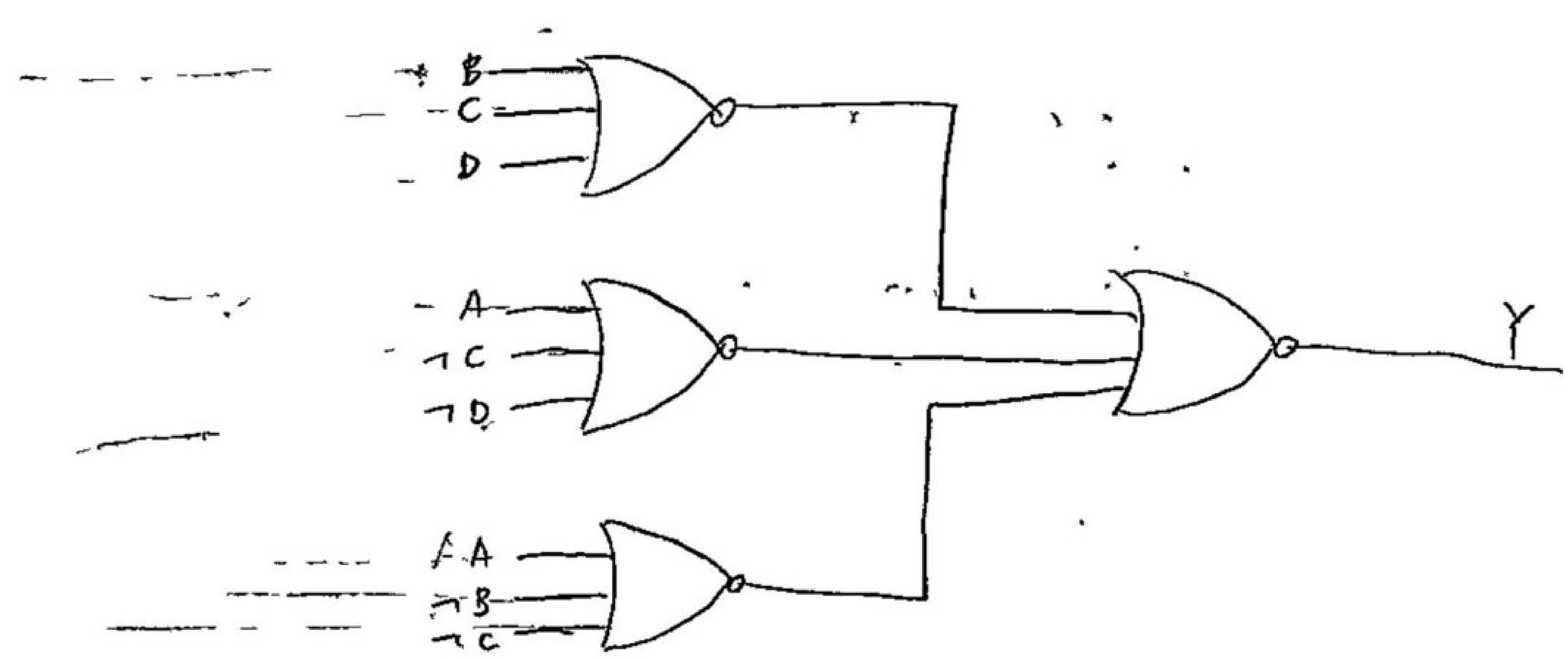
EssentialPrimeImplicants = $\neg A \wedge B \wedge C$

(d) Implement the function *Y* (not ~Y) using a tree of 2 input multiplexers as shown. The select signals are *A*, *B*, and *C*. The top input of the multiplexer is selected when the select-signal=0 and vice versa. Write the desired inputs on the figure below. You may use *D* or ~*D* as input but avoid using them as much as possible.



(e) Write Boolean expression for Y as a *minimal* product-of-sum. Then implement by using only NOR gates (each gate can have multiple inputs). Use the minimum number of NORs with fewest # of total inputs (minimize literals and terms). You may assume true and complement inputs are available.

$Y = (\neg B \vee C \vee D) \wedge (A \vee \neg C \vee \neg D) \wedge (A \vee \neg B \vee \neg C)$



Question #2

$$Y = \neg(\neg a \wedge d) \vee (\neg e \wedge \neg(c \wedge b))$$

(a) For the above Boolean function, convert the above expression into a minimal product-of-sum.

$Y = (a \vee \neg d) \vee (\neg e \wedge (\neg b \vee \neg c))$

$= a \vee \neg d \vee (\neg b \wedge \neg e) \vee (\neg e \wedge \neg c)$

$Y =$ $a \vee (\neg b \wedge \neg e) \vee (\neg c \wedge \neg e) \vee \neg d$

(b) If you need to represent the equation in (a) in a Fully **Conjunctive** Normal Form, how many terms, and write the expression.

$a \vee ((\neg b \vee \neg c) \wedge (\neg b \vee \neg e) \wedge (\neg c \vee \neg e) \wedge \neg e) \vee \neg d$

#	a	b	c	d	e	Y	
0	0	0	0	0	0	1	
1	0	0	0	0	1	1	
2	0	0	0	1	0	1	
3	0	0	0	1	1	0	← $a \vee b \vee c \vee \neg d \vee \neg e$
4	0	0	1	0	0	1	
5	0	0	1	0	1	1	
6	0	0	1	1	0	1	
7	0	0	1	1	1	0	← $a \vee b \vee \neg c \vee \neg d \vee \neg e$
8	0	1	0	0	0	1	
9	0	1	0	0	1	1	
10	0	1	0	1	0	1	
11	0	1	0	1	1	0	← $a \vee \neg b \vee c \vee \neg d \vee \neg e$
12	0	1	1	0	0	1	
13	0	1	1	0	1	1	
14	0	1	1	1	0	0	← $a \vee \neg b \vee \neg c \vee \neg d \vee e$
15	0	1	1	1	1	0	← $a \vee \neg b \vee \neg c \vee \neg d \vee \neg e$
	1	x	x	x	x	1	

$a \vee \neg b \vee \neg c \vee \neg d \vee e$
$a \vee \neg b \vee \neg c \vee \neg d \vee \neg e$
$a \vee \neg b \vee c \vee \neg d \vee \neg e$
$a \vee b \vee \neg c \vee \neg d \vee \neg e$
$a \vee b \vee c \vee \neg d \vee \neg e$

Number of sum terms: 5

$\prod M(3, 7, 11, 14, 15)$

$Y =$ $(a \vee b \vee c \vee \neg d \vee \neg e) \wedge (a \vee b \vee \neg c \vee \neg d \vee \neg e) \wedge (a \vee \neg b \vee c \vee \neg d \vee \neg e) \wedge (a \vee \neg b \vee \neg c \vee \neg d \vee e) \wedge (a \vee \neg b \vee \neg c \vee \neg d \vee \neg e)$

Question #3

The following 10-b word can be used to represent different numbers depending on the encoding
10b'11_1001_0001

(a) If the word is 2's complement, what is the corresponding integer? -111

$-2^9 + 2^8 + 2^7 + 2^4 + 2^0 = -111$

(b) If the word is unsigned fixed point 5.5, what is the corresponding number?
28.53125 $2^4 + 2^3 + 2^2 + 2^{-1} + 2^{-5} = 28.53120$

What is the absolute accuracy of this representation? $2^{-6} = 0.015625$

(c) What is this word in Binary Coded Decimal? 391

$2^1 + 2^0 = 3$
$2^3 + 2^0 = 9$
$2^0 = 1$

(d) If the word is 4E5 floating point number (IEEE format S+EEEEE+MMMM),

What is the bias? 15 $2^{5-1} - 1 = 15$

1_11001_0001 What is the corresponding real number? -1088

$M = 1.0001_2 = 1 + 2^{-4} = 1.0625$
$E = exp - bias = (2^4 + 2^3 + 2^0) - 15 = 10$.
$S = 1$. $\boxed{-1.0625 \times 2^{10}}$

What is the relative accuracy as a % of this representation? 3.125%

$2^{-mant\ bits} = 2$ (rel acc) rel acc $= 2^{-1-4} = 2^{-5} = 3.125\%$

(e) In the IEEE 754 floating point representation, what is the advantage of the implied 1 for the mantissa?

- Allowing a wider range of mantissa, and allowing only one unique representation for a given real number.

What is the advantage of using a Bias in the exponent?

Allowing both very large magnitudes and very small magnitudes (where $E < 0$).

(f) In a binary 2's complement system, how do you determine if a number is negative?

Check if the most significant bit is 1.

What is the logic for determining an underflow during an subtraction between 2 input words of n-bits, a[n-1:0] and b[n-1:0]?

If the borrow out bit of the subtractor is 1, an underflow occurred.

(g) In base-5 system and using 3 "quinits" (base-5 digits) in 5's complement:

How would one represent a base-10 integer -55? 240

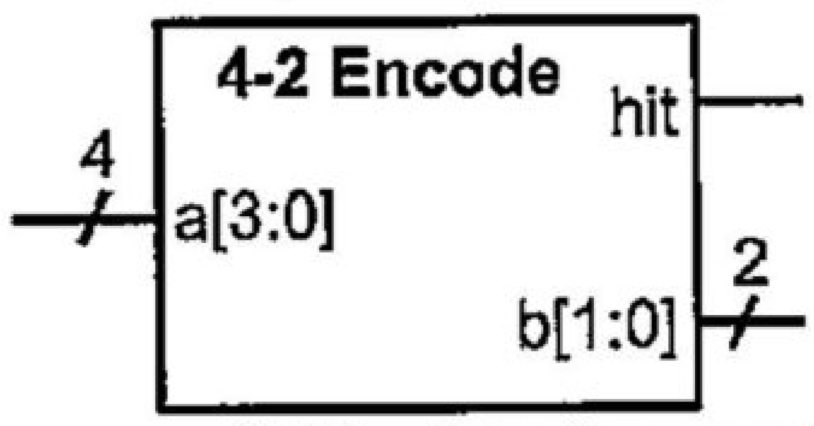
representation:
$5^3 - 55 = 10_{10} =$
$2\times5^2 + 4\times5^1 + 0\times5^0$
$= 240_5$

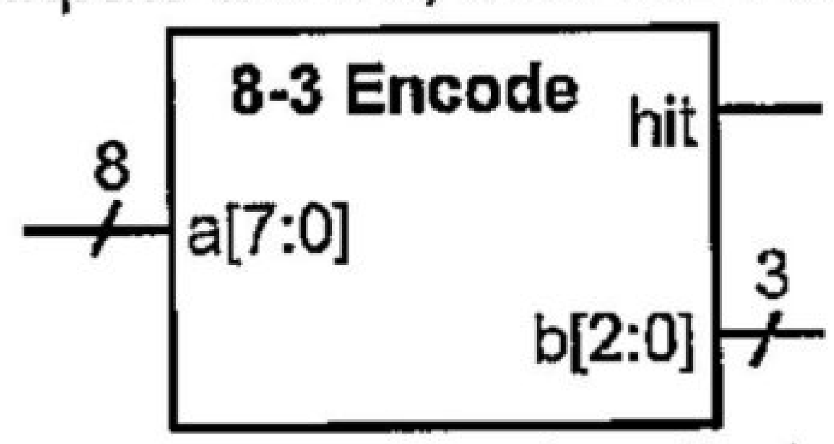
What is the most positive value in base-10 integer that can be represented?
62

$\left\lfloor \frac{5^3 - 1}{2} \right\rfloor = 62_{10}$

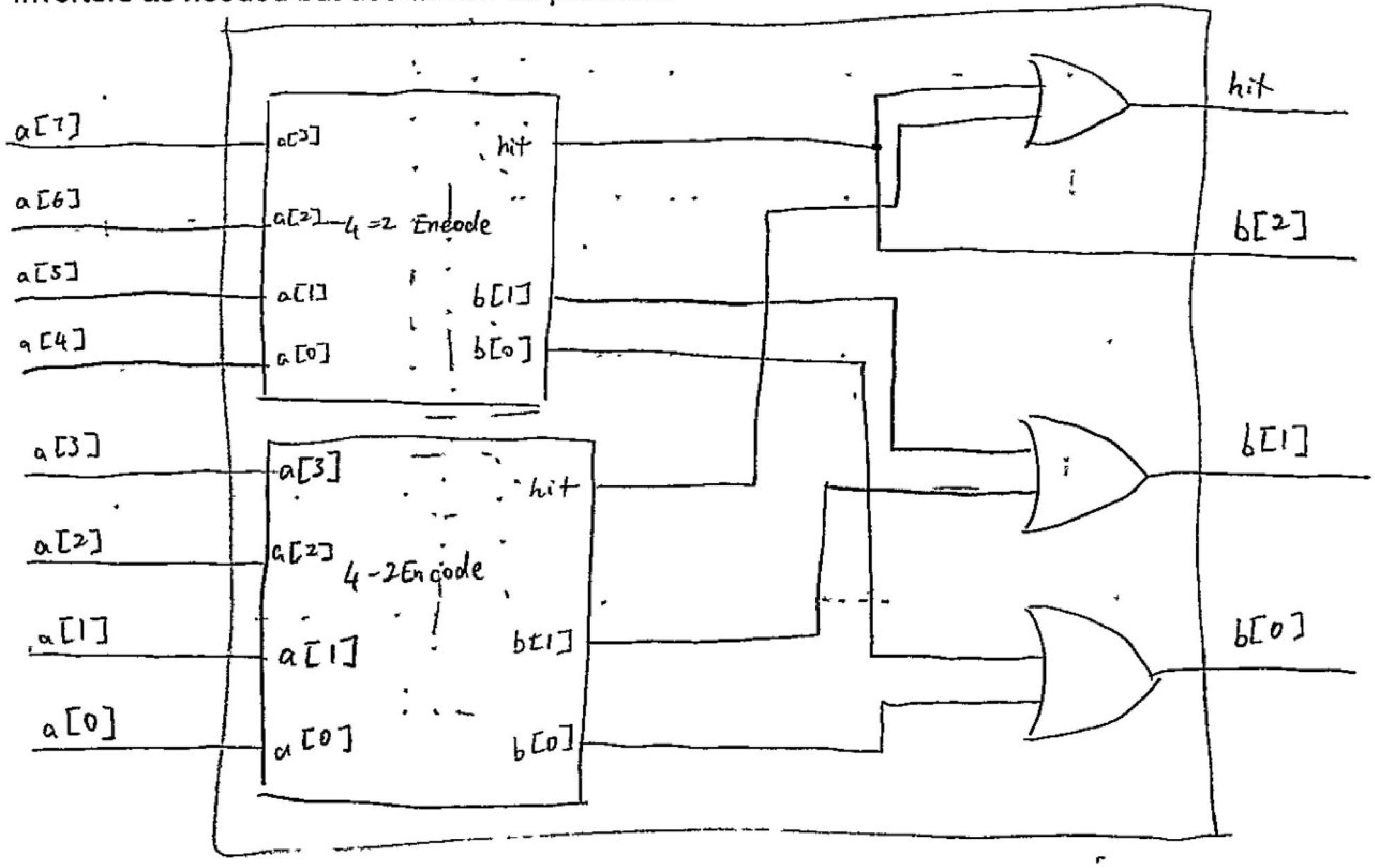
Question #4

A 4-to-2 encoder (from one-hot to binary) is shown as a block below. The inputs are 4 one-hot inputs, *a[3:0]*. The output are 2 binary bits, *b[1:0]*, indicating the position of the "hot" input bit. If the input has a hot bit, then *hit*=1'b1, otherwise, when all inputs are 0's, then *hit*=1'b0.

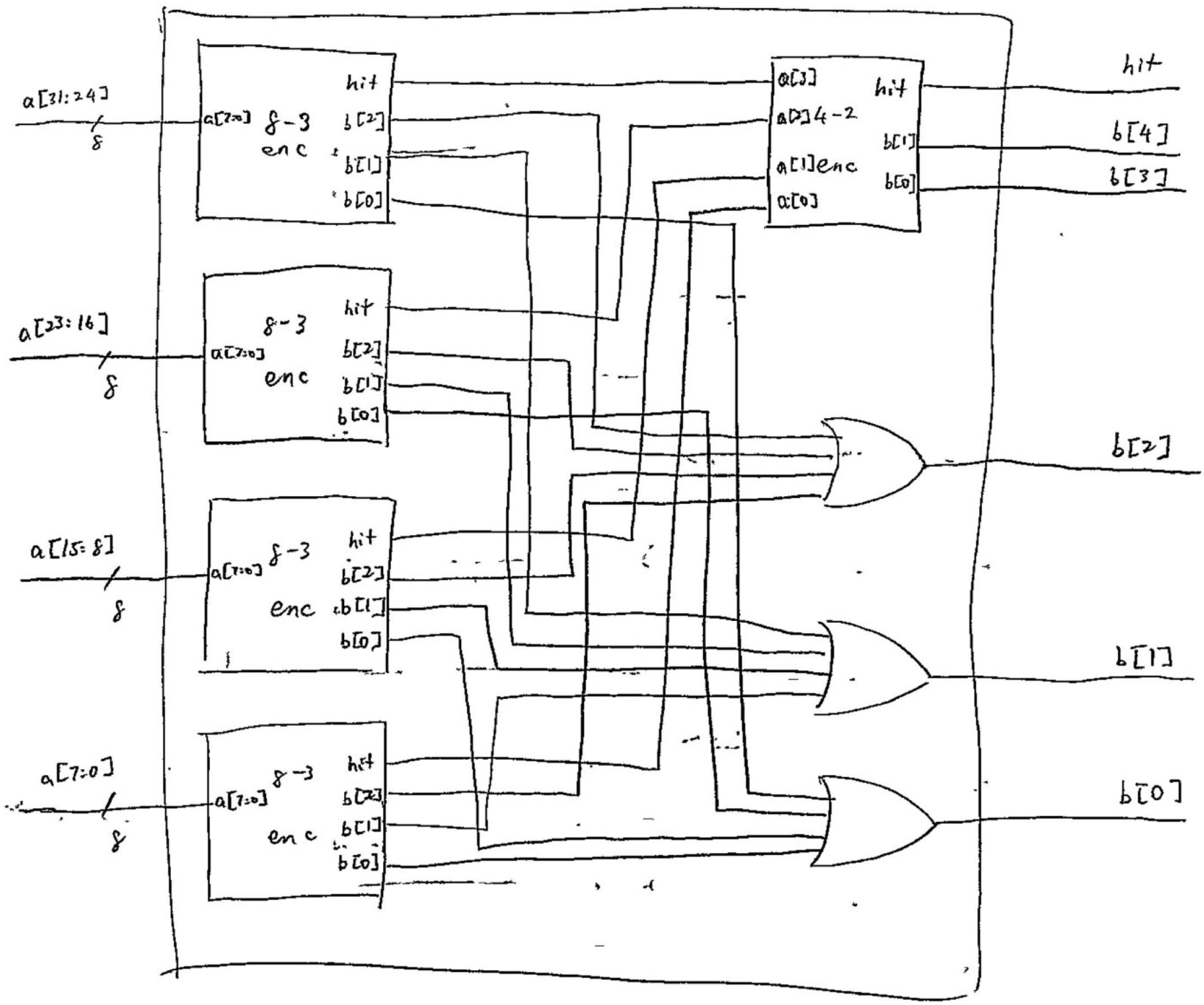




(a) Use instances of this 4-to-2 encoder block to design an 8-to-3 encoder. On-hot inputs are a[7:0], binary outputs are b[2:0], and a *hit* indicator. You may use additional OR, AND, or Inverters as needed but use as few as possible.

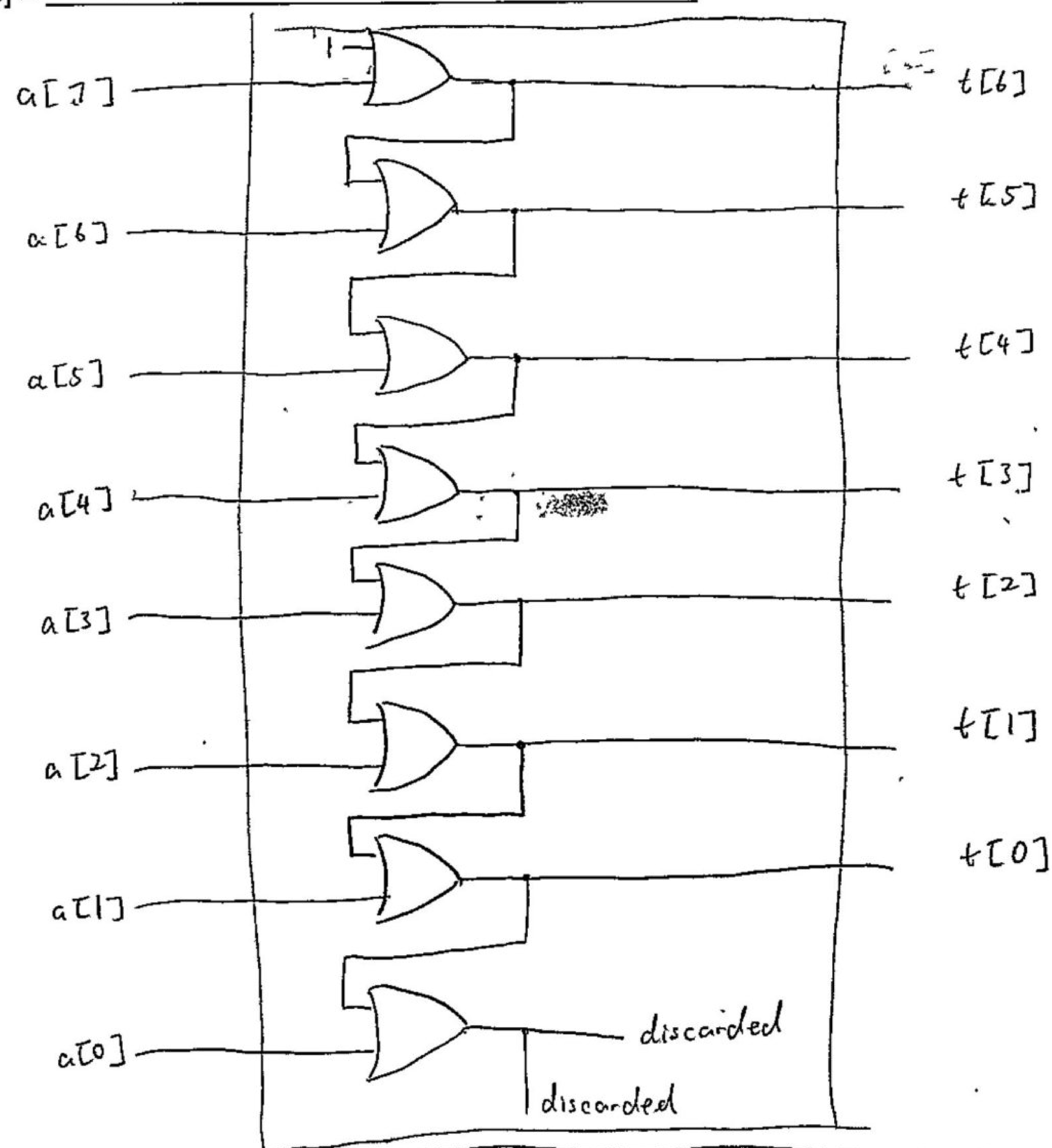


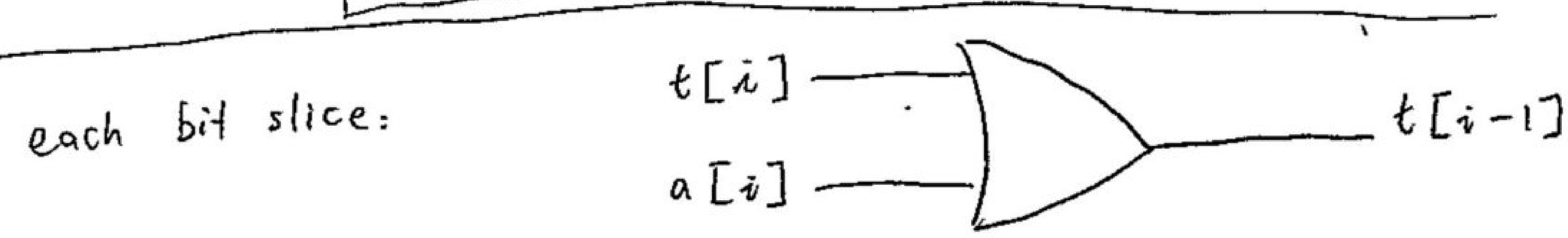
(b) Use 8-to-3 encoders and 4-to-2 encoders as modules and design a 32-5 encoder with as few modules and additional logic as possible. Again, you may use additional OR, AND, or Inverters as needed.



(c) Instead of an encoder that converts to binary, consider a converter that outputs thermometer code instead. For instance, an 8-bit one-hot input, *a[7:0]*, would convert to 7-bit thermometer output, *t[6:0]*. When *a[7:0]* = 8'0100_0000, *t[6:0]* = 7'b011_1111, and when *a[7:0]* = 8'b0000_0001, *t[6:0]* = 7'b000_0000. First write the Boolean expression for *t[3]* as a function of the a[7:0] inputs. Then design, using a **bit-cell approach**, the logic for each bit position. *Denote the inputs, outputs and signals passing between bit-positions clearly as well as the connections for the MSB and LSB positions.* Note that since the output has one fewer bit than the input, one of the bit-slices may have logic that is not used to produce an output. You may only use OR, AND, or Inverters as needed but use as few as possible.

t[3] = a[7] ∨ a[6] ∨ a[5] ∨ a[4]





Blank page for scratch work

Blank page for scratch work