CS2100 Tutorial 7

Combinatorial Circuits

Slides by Theodore, adapted from Prof. Aaron's slides

Announcements (Tutorial 7)

- Midterm: results out, please check
  - Any queries ask prof Weng Fai / Anandha
- Assignment 2: done
  - Take your marked paper, Canvas results out on Sunday
- Assignment 1: last call
  - According to my notes my backlog are done, emails sent
  - Last call if I miss anything (Q5, Q1+4+Admin)
- Deepavali / Well-being day (in 2 week)
  - Mass makeup tutorial (zoom) on Wednesday, auto-attendance
- Will start at :05 as usual

Recap

- Adders
- Comparators

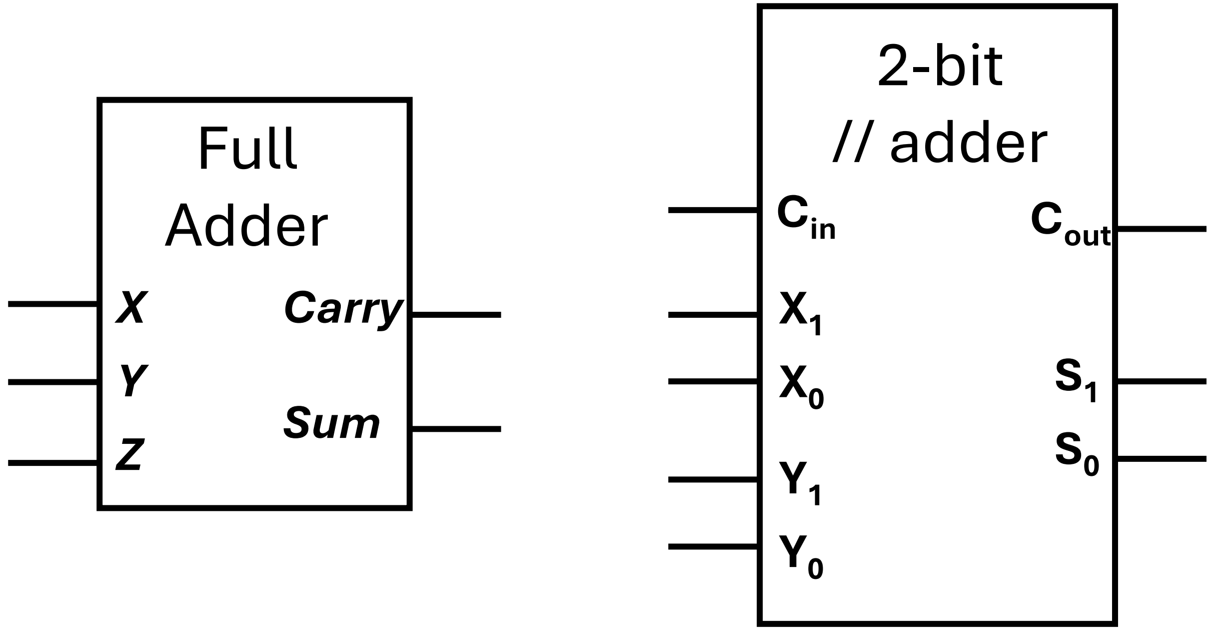
D3. A combinational circuit takes in a 5-bit input *ABCDE* and generates a 2-bit value *PQ* such that *PQ* represents the *distance* between the two closest 1s in the input. The distance is defined to be the number of 0s between the two closest 1s.

You may assume that the distance is always determinable from the given input. Therefore, inputs such as 00000 and 01000 will not be supplied to this circuit.

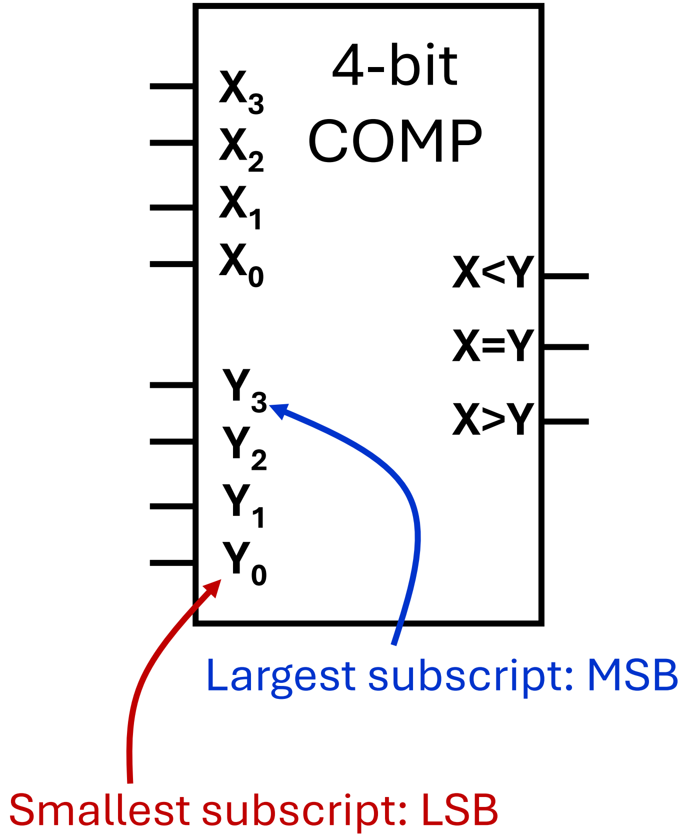
Q1. You are to design a circuit to implement a function *V*(*A*,*B*,*C*,*D*,*E*) that takes in input *ABCDE* and generates output 1 if *ABCDE* is a valid input for the circuit in question D3 above, or 0 if *ABCDE* is an invalid input.

Q1.

Idea?



Full Adder
X
Y
Z
Carry
Sum
2-bit // adder
C_in
X_1
X_0
Y_1
Y_0
C_out
S_1
S_0



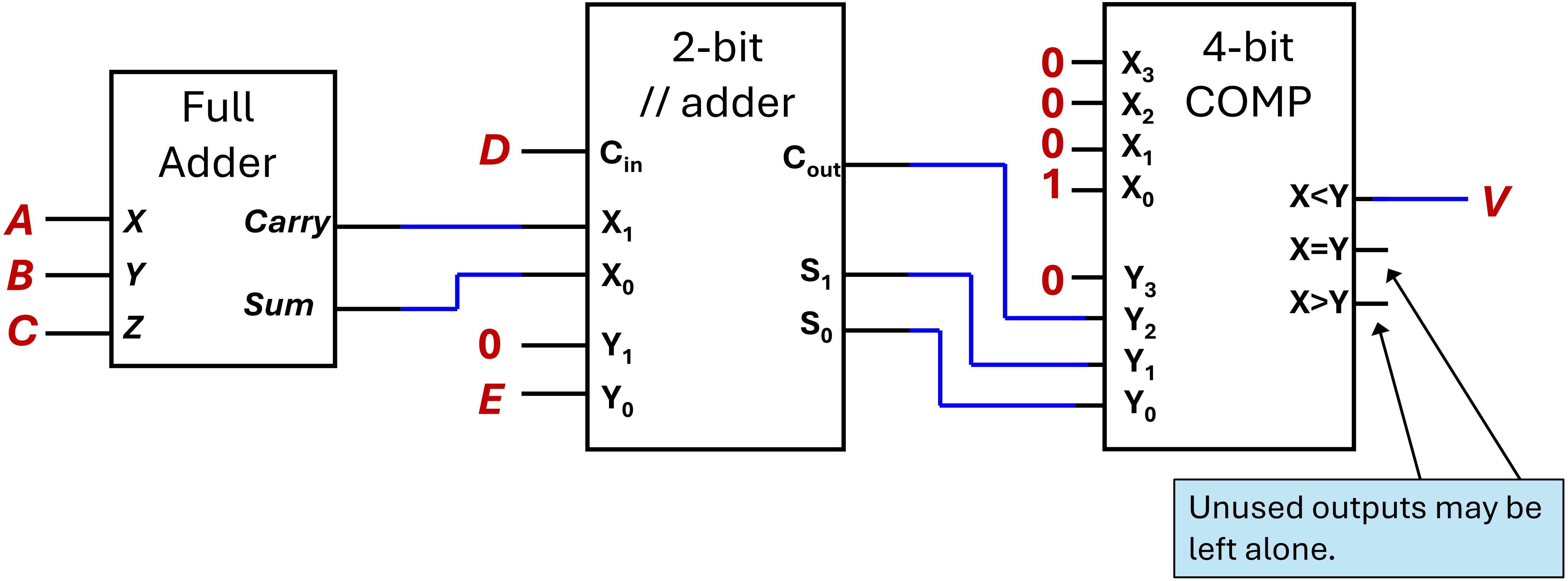
4-bit COMP
X_3
X_2
X_1
X_0
Y_3
Y_2
Y_1
Y_0
X<Y
X=Y
X>Y
Largest subscript: MSB
Smallest subscript: LSB

Q1.

Idea?

Count the number of 1's in *ABCDE*.
If > 1 then $V = 1$, else $V = 0$.

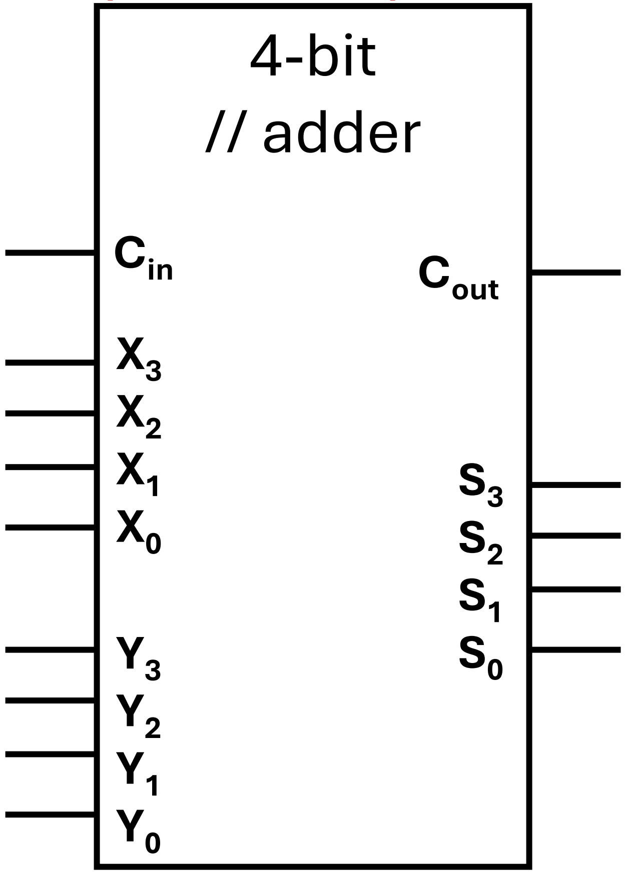
Important: All inputs must be connected to some value! Leave no inputs "hanging".



Q2(a)

A	*B*	*C*	*D*	*E*	*F*	*G*	*H*
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	0	0	1				
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

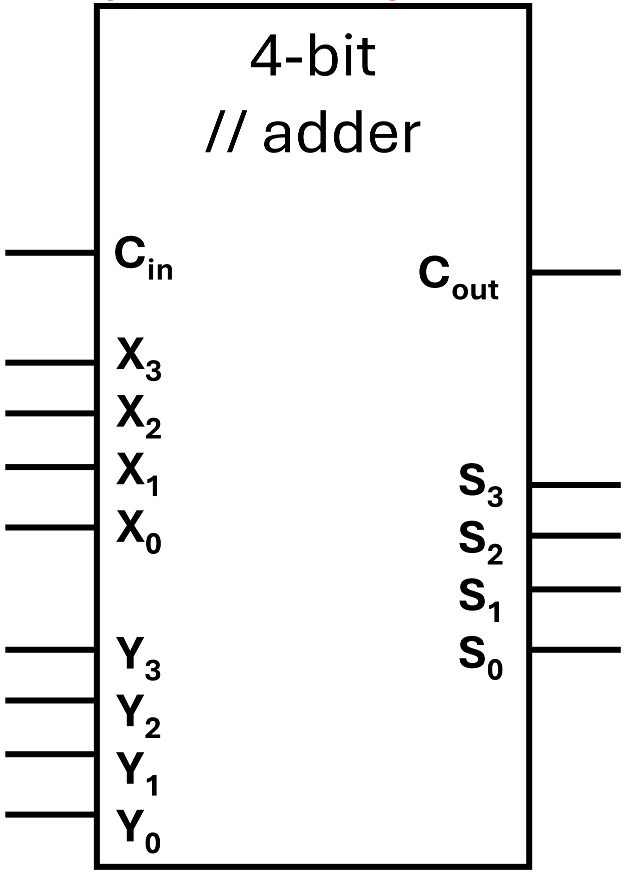
EFGH = (*ABCD*+1)/2



Q2(a)

A	*B*	*C*	*D*	*E*	*F*	*G*	*H*
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	0	1
0	0	1	1	0	0	1	0
0	1	0	0	0	0	1	0
0	1	0	1	0	0	1	1
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	0
1	0	0	1	0	1	0	1
1	0	1	0	0	1	0	1
1	0	1	1	0	1	1	0
1	1	0	0	0	1	1	0
1	1	0	1	0	1	1	1
1	1	1	0	0	1	1	1
1	1	1	1	1	0	0	0

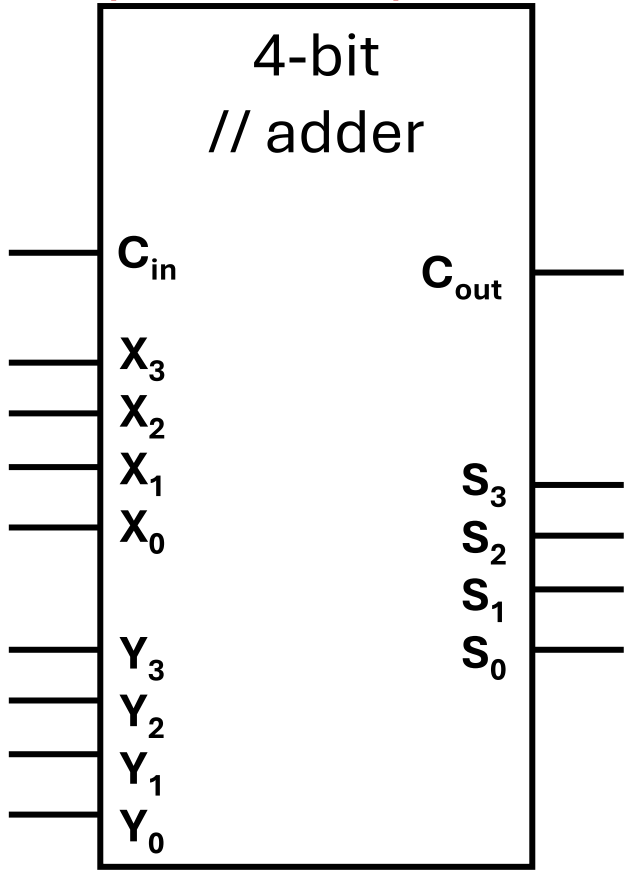
EFGH = (*ABCD*+1)/2



Q2(a)

A	B	C	D	E	F	G	H
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	0	1
0	0	1	1	0	0	1	0
0	1	0	0	0	0	1	0
0	1	0	1	0	0	1	1
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	0
1	0	0	1	0	1	0	1
1	0	1	0	0	1	0	1
1	0	1	1	0	1	1	0
1	1	0	0	0	1	1	0
1	1	0	1	0	1	1	1
1	1	1	0	0	1	1	1
1	1	1	1	1	0	0	0

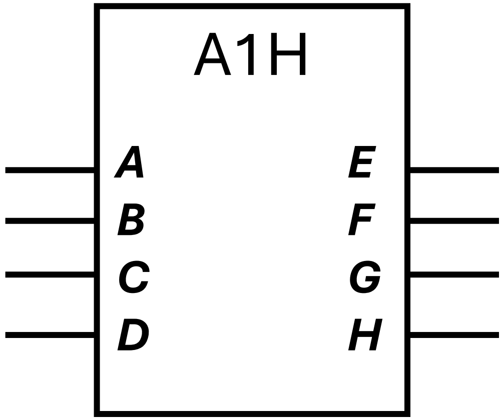
EFGH = (*ABCD*+1)/2



Q2(b)

4221-to-8421 decimal code converter

P	*Q*	*R*	*S*	*W*	*X*	*Y*	*Z*
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	1	0	0	1	0	0
1	0	0	1	0	1	0	1
1	1	0	0	0	1	1	0
1	1	0	1	0	1	1	1
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	1



Comparison of 8421 and 4221 code

8421 code (BCD)

- 4 bits for every digit
  - Weights for each digits: 8, 4, 2, 1
  - e.g. $7 = (8\times0)+(4\times1)+(2\times1)+(1\times1)$ represented as 0111
- WXYZ means $8\times W + 4\times X + 2\times Y + 1\times Z$

4221 code

- 4 bits for every digit
  - Weights for each digits: 4, 2, 2, 1
  - e.g. $7 = (4\times1)+(2\times1)+(2\times0)+(1\times1)$ represented as 1101
- PQRS means $4\times P + 2\times Q + 2\times R + 1\times S$

Converting 4221 to 8421 code

PQRS means $4{\times}P + 2{\times}Q + 2{\times}R + 1{\times}S$
WXYZ means $8{\times}W + 4{\times}X + 2{\times}Y + 1{\times}Z$

- S needs to go to Z; as the only representation of 1
- We now have

$$4(P) + 2(Q) + 2(R) = 8(W) + 4(X) + 2(Y)$$
$$2(P) + Q + R = 4(W) + 2(X) + Y$$

which means that PQ+R = WXY, which is the (second) solution to the A1H (see slide 9 bottom right corner).

Q3

BCD code

Digits:	0	1	2	3	4	5	6	7	8	9
Code:	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Given two decimal digits A and B, represented by their BCD codes $A_3A_2A_1A_0$ and $B_3B_2B_1B_0$ respectively, implement a circuit without using any logic gates to calculate the BCD code of the 3-digit output of $(51\times A) + (20\times(B\%2))$, where % is the modulo operator. Name the outputs $F_{11}F_{10}F_9F_8\ F_7F_6F_5F_4\ F_3F_2F_1F_0$.

Hint: Fill in the table on the right.

A				**5×*A***							
A_3	A_2	A_1	A_0								

Q3

BCD code

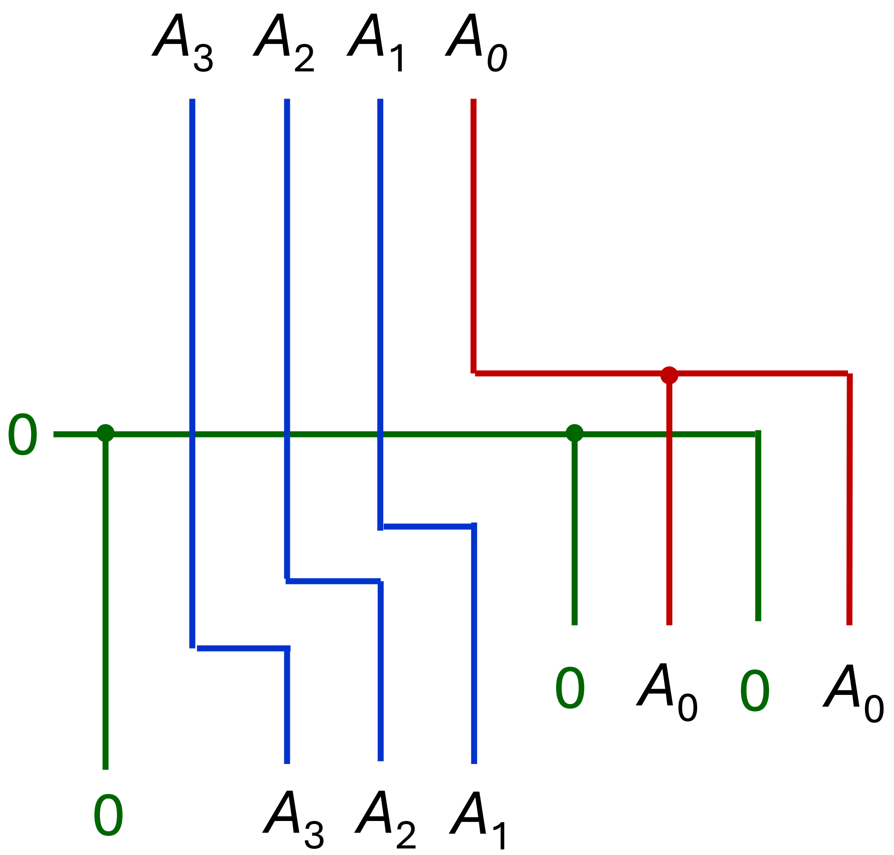
Digits:	0	1	2	3	4	5	6	7	8	9
Code:	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

A				5×*A*							
A_3	A_2	A_1	A_0								
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	1	0	1
0	0	1	0	0	0	0	1	0	0	0	0
0	0	1	1	0	0	0	1	0	1	0	1
0	1	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	1	0	0	1	0	1
0	1	1	0	0	0	1	1	0	0	0	0
0	1	1	1	0	0	1	1	0	1	0	1
1	0	0	0	0	1	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	1	0	1

Q3

BCD code

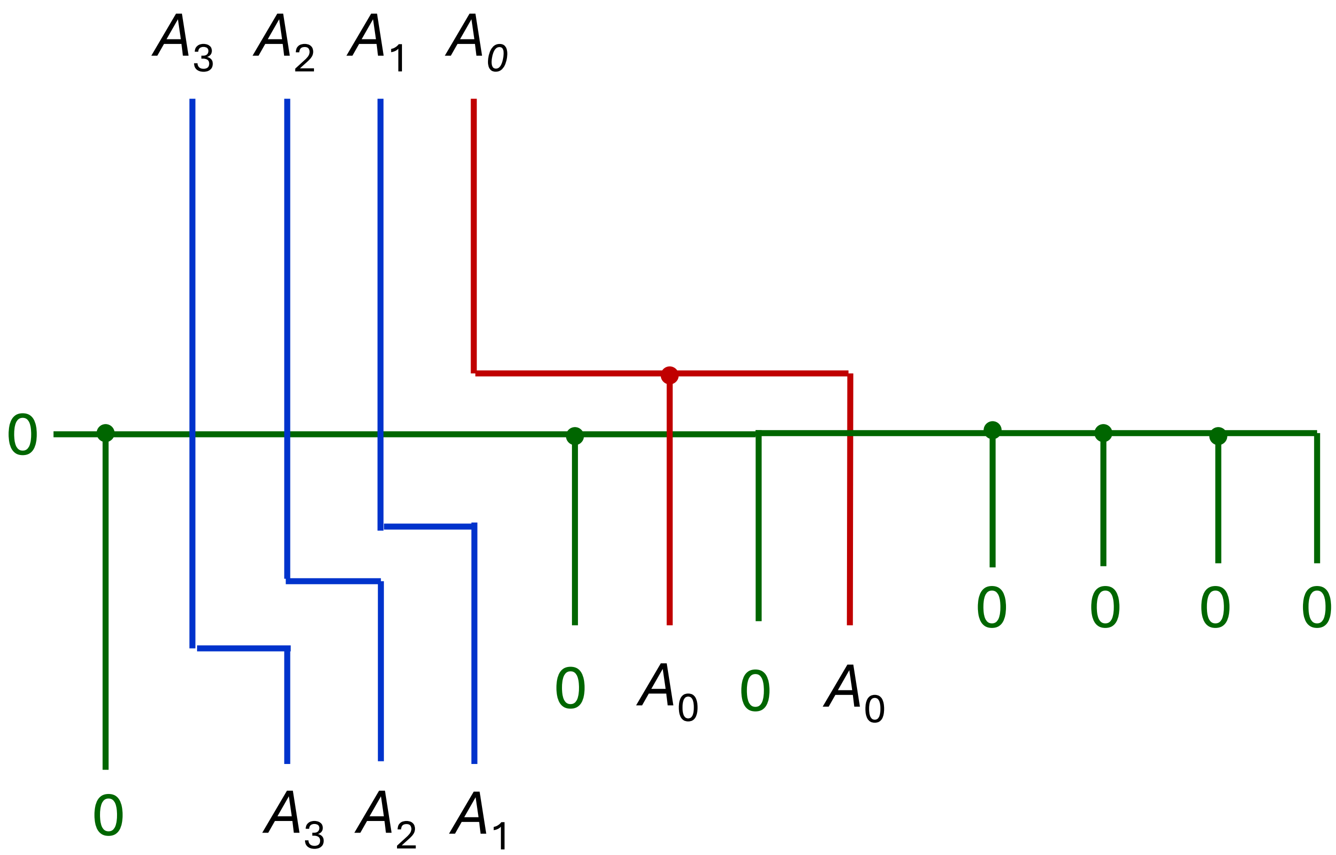
Digits:	0	1	2	3	4	5	6	7	8	9
Code:	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001



A				$5\times A$							
A_3	A_2	A_1	A_0								
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	1	0	1
0	0	1	0	0	0	0	1	0	0	0	0
0	0	1	1	0	0	0	1	0	1	0	1
0	1	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	1	0	0	1	0	1
0	1	1	0	0	0	1	1	0	0	0	0
0	1	1	1	0	0	1	1	0	1	0	1
1	0	0	0	0	1	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	1	0	1

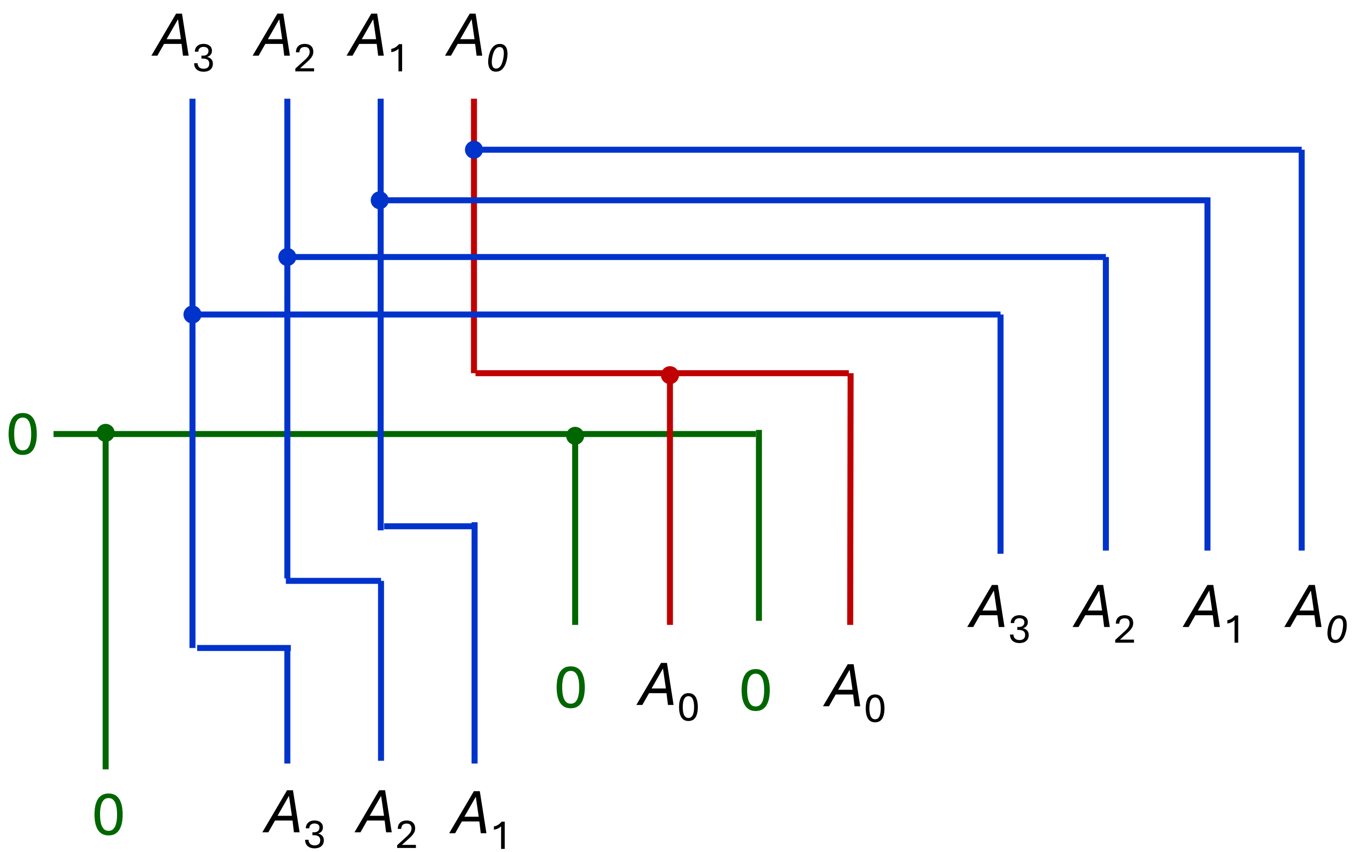
Q3

$\cancel{5 \times A}$ $\cancel{50 \times A}$ $51 \times A$



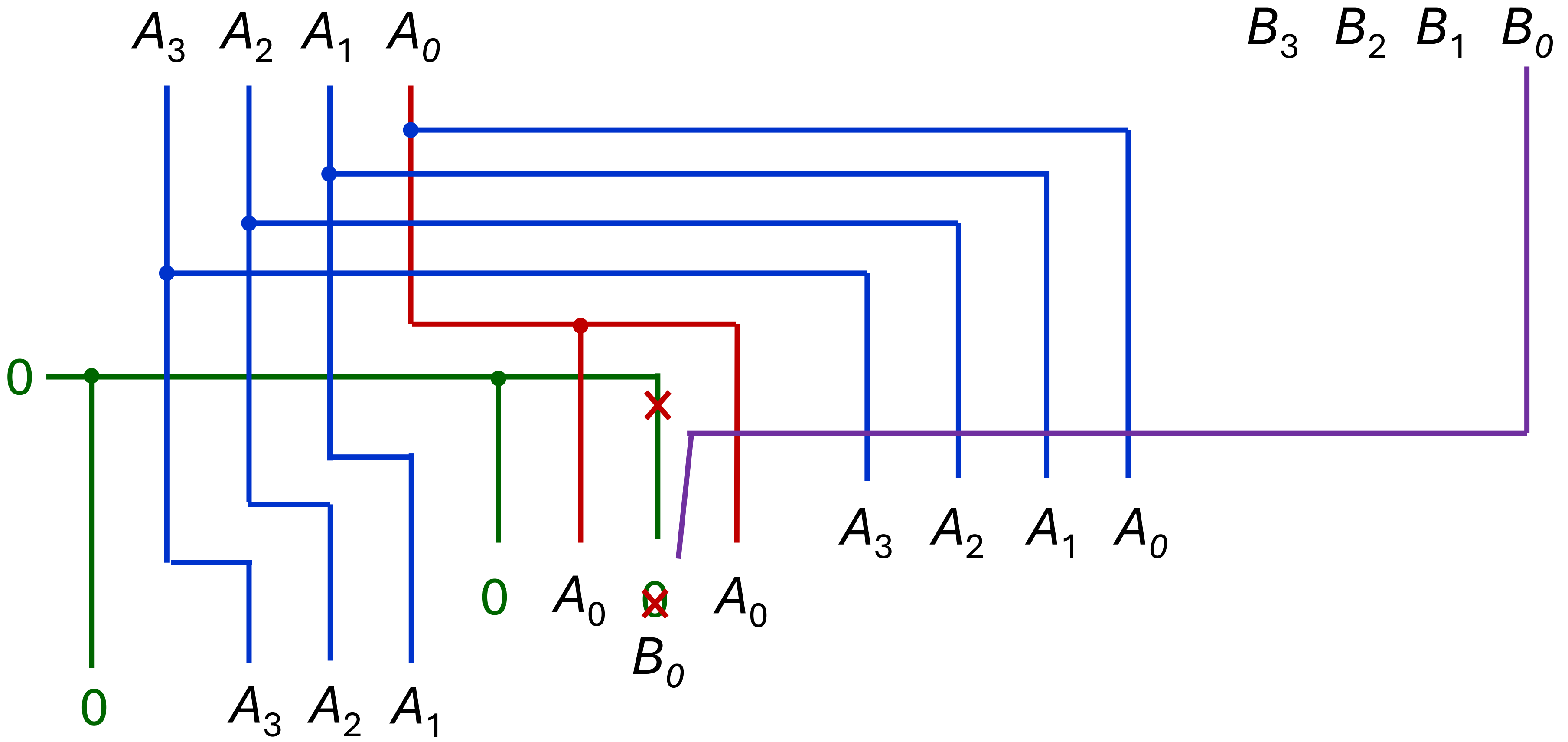
Q3

$\cancel{5\times A}$ $\cancel{50\times A}$ $\cancel{51\times A}$ $51\times A + (20\times(B\%2))$



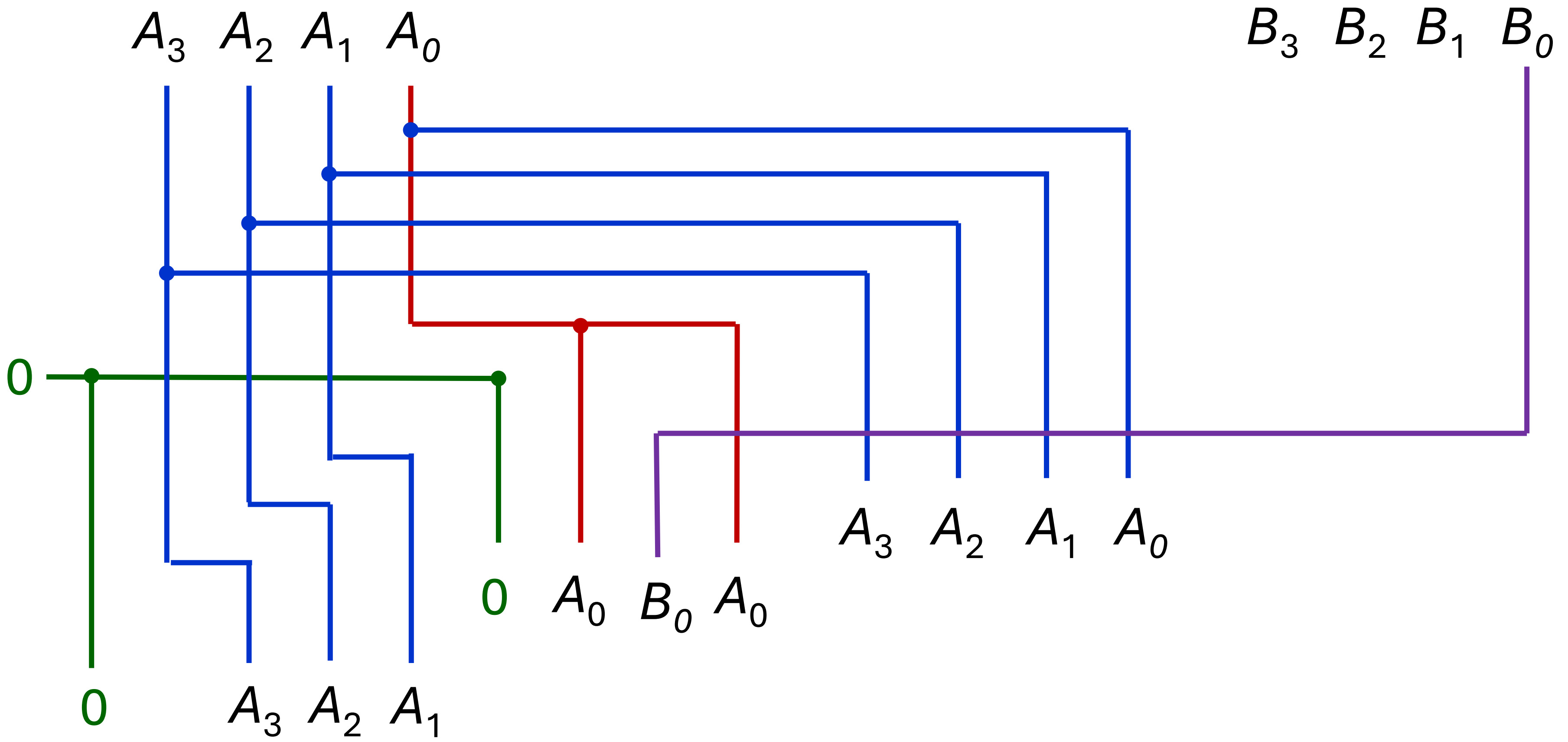
Q3

~~$5\times A$~~ ~~$50\times A$~~ ~~$51\times A$~~ $51\times A + (20\times(B\%2))$

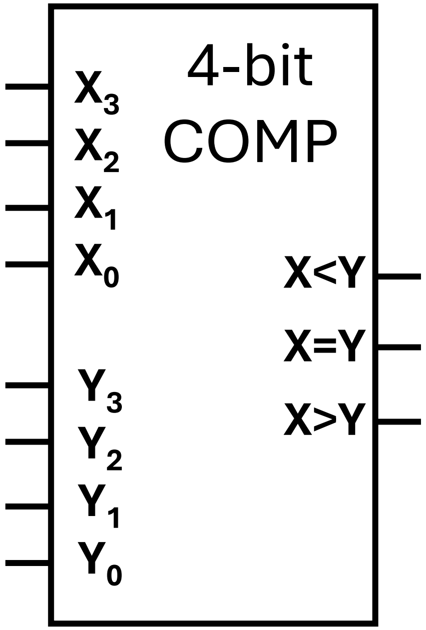


Q3

$\cancel{5\times A}$ $\cancel{50\times A}$ $\cancel{51\times A}$ $51\times A + (20\times(B\%2))$



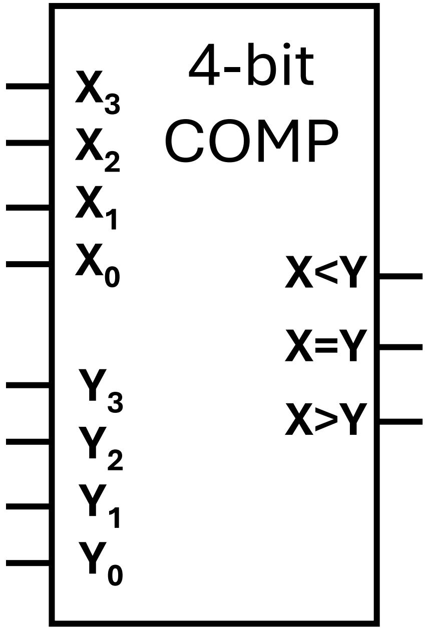
Q4(a) $F(A,B,C,D) = \Sigma m(12 - 15)$.



Q4(b) $G(A,B,C,D) = \Sigma m(0, 6, 9, 15)$.

A	B	C	D	G
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

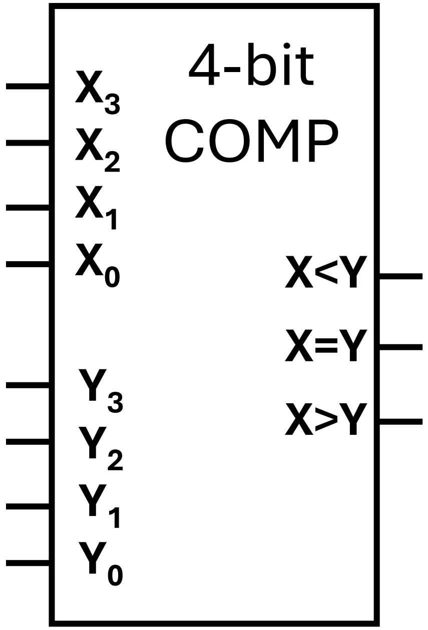
Observe the pattern of *ABCD* where *G*=1.



Q4(c) $H(A,B,C,D) = \Sigma m(0, 1, 6, 7, 8, 9, 14, 15)$.

A	*B*	*C*	*D*	*H*
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

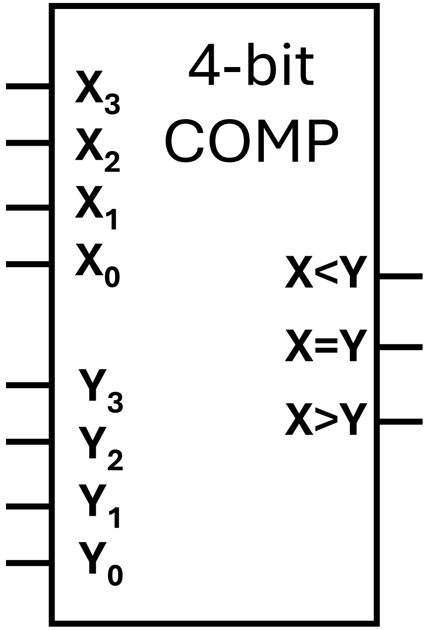
Observe the pattern of *ABCD* where *H*=1.



Q4(d) $Z(A,B,C,D) = \Sigma m(1, 3, 5, 7, 9, 11, 13)$.

A	*B*	*C*	*D*	*Z*
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

Tough!



If this is 1 then it becomes very easy.

End of Tutorial 7

- Slides uploaded on github.com/theodoreleebrant/TA-2425S1
- Email: theo@comp.nus.edu.sg
- Anonymous feedback:
  bit.ly/feedback-theodore
  (or scan on the right)

(Also reminder for me to take attendance)