

CSE140 HW5 Preparation

Xinyuan Wang

06/01/2018

Q1: Subtractor

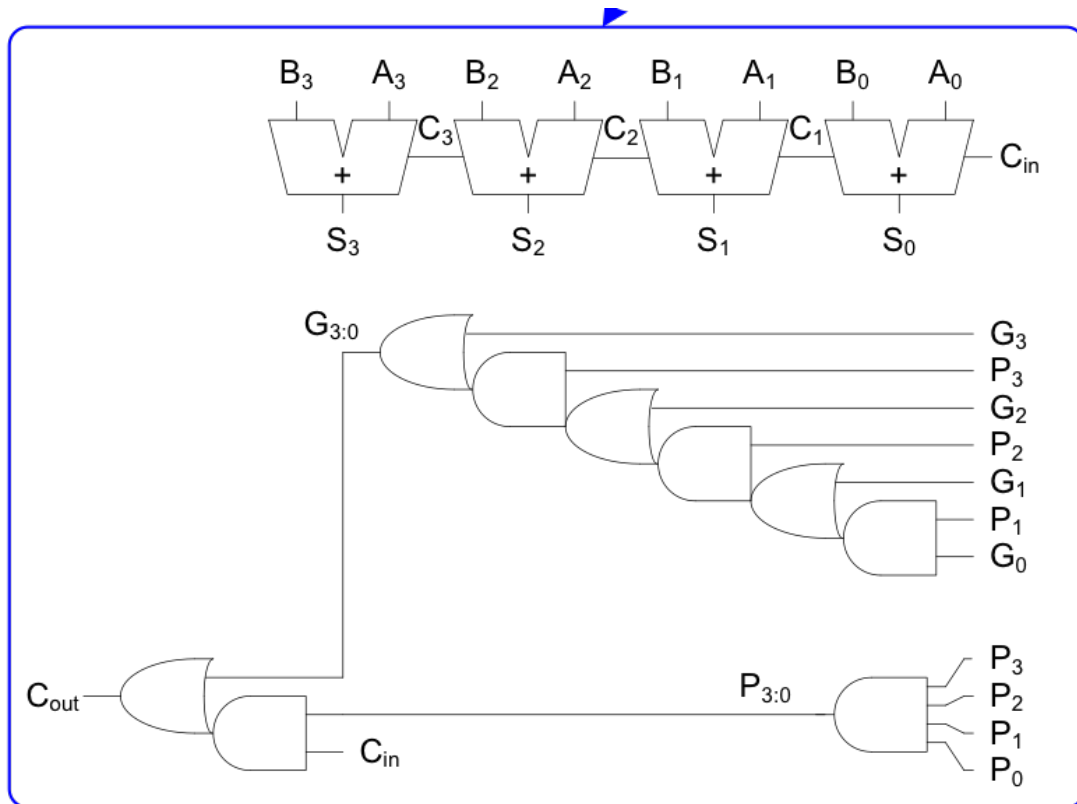
- How we implement the subtraction between binary bits?
 - ✓ Given 2 **binary** inputs x, y and borrow in bit , calculate $x - y$

$$\begin{array}{rcccc} \text{borrow out} & & & & \\ & 1 & 0 & 0 & \\ & 1 & 0 & 1 & x \\ - & 1 & 1 & 1 & y \\ \hline & 1 & 1 & 0 & \end{array}$$

- ✓ A full subtractor with input $(x, y, b_{in}) \Rightarrow$ output (b_{out}, d)
- ✓ Write the truth table and derive Boolean expressions

Q1: Subtractor

- A four bit borrow look ahead subtractor
 - ✓ Refer to the carry look ahead adder



- A N-bit carry look ahead adder is usually much faster than ripple than a ripple carry adder if $N > 16$
 - ✓ Delays for carry look ahead and ripple carry adder are defined in slide 18, 23 of lecture 14

Q1: Subtractor

- A four bit borrow look ahead subtractor
 - ✓ Use full subtractors and minimal logic gates
 - ✓ Generate G_i and propagate P_i signals for each column
 - ❖ When a column will generate a borrow out?
 - ❖ When a column will propagate a borrow in to borrow out?

Q2: Prefix Adder

- Compute the carry in for each column as fast as possible
- Has $\log_2(N)$ stages
- Column -1 holds C_{in} , so $G_{-1} = C_{in}, P_{-1} = 0$
- Compute the **prefixes**: $G_{0:-1}, G_{1:-1}, G_{2:-1}, \dots, G_{N-1:-1}$
 - ✓ the generate and propagate for a block spanning bits $i:j$

$$G_{i:j} = G_{i:k} + P_{i:k}G_{k-1:j}$$

$$P_{i:j} = P_{i:k}P_{k-1:j}$$

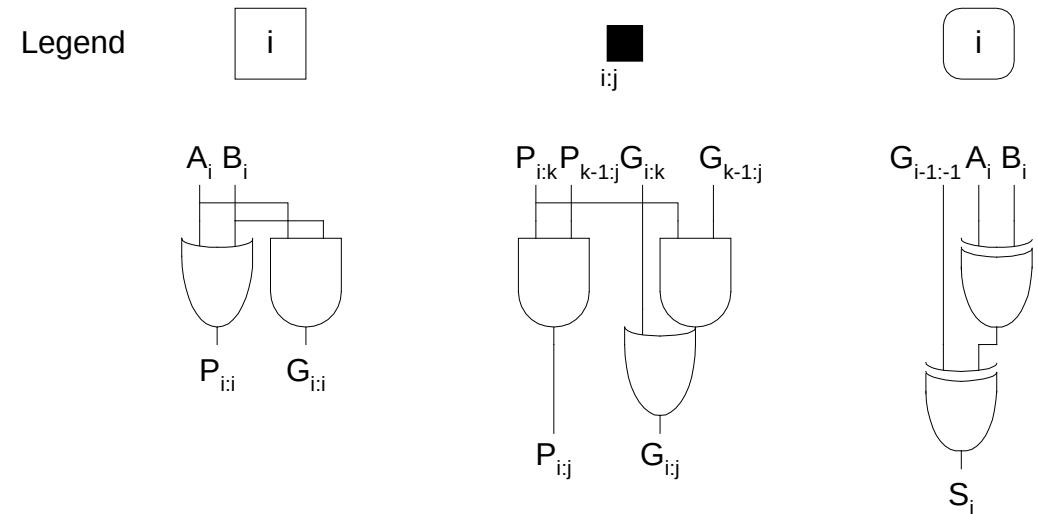
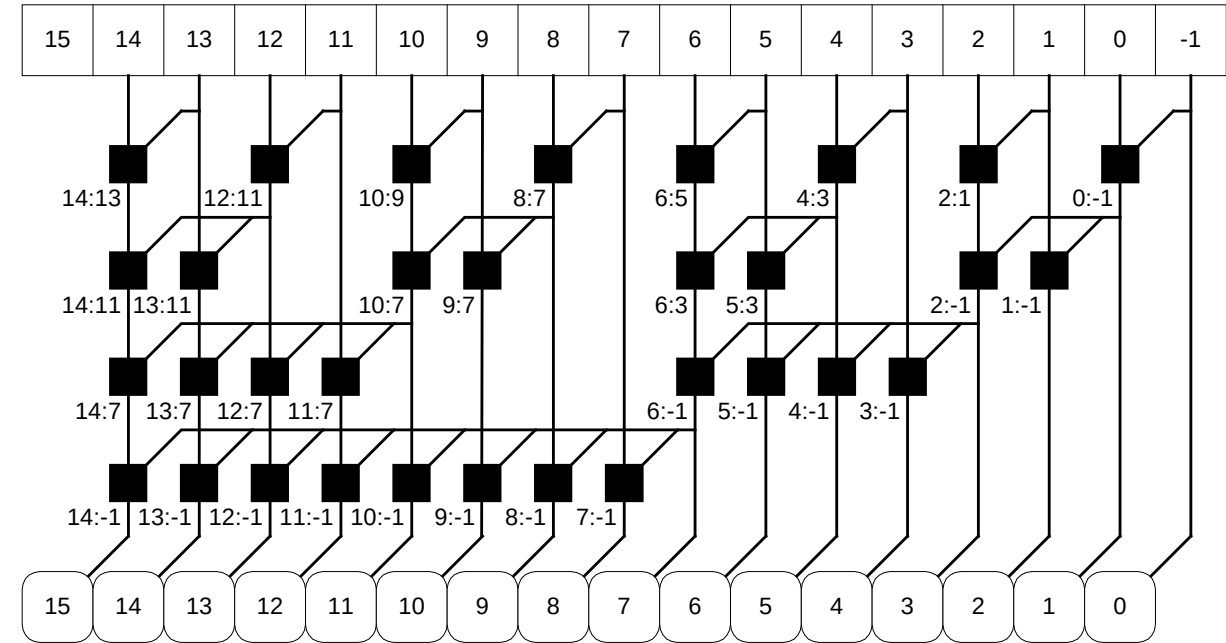
Note: $G_{i:i} = A_iB_i, P_{i:i} = A_i + B_i$ are the generate and propagate for column i

- The sum of each column

$$S_i = (A_i \oplus B_i) \oplus G_{i-1:-1}$$

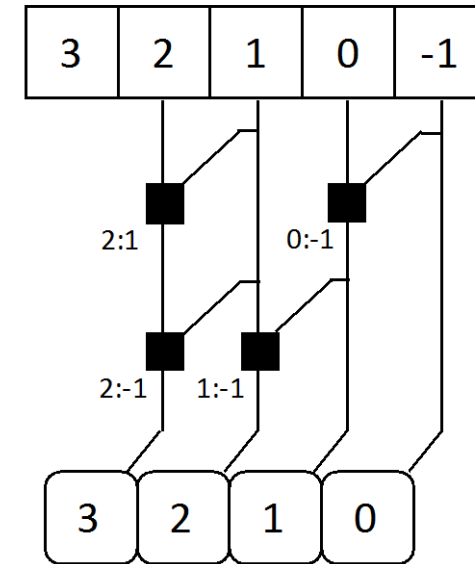
Q2: Prefix Adder

- A 16-bit Prefix Adder from slide 27 of Lecture 14
 - ✓ Has 4 stages
 - ✓ Compute $(G_{i:i}, P_{i:i})$ for each column
 - ✓ Compute $(G_{i:j}, P_{i:j})$ for each box
 - ✓ Compute sum S_i with $A_i, B_i, G_{i-1:-1}$ for each column

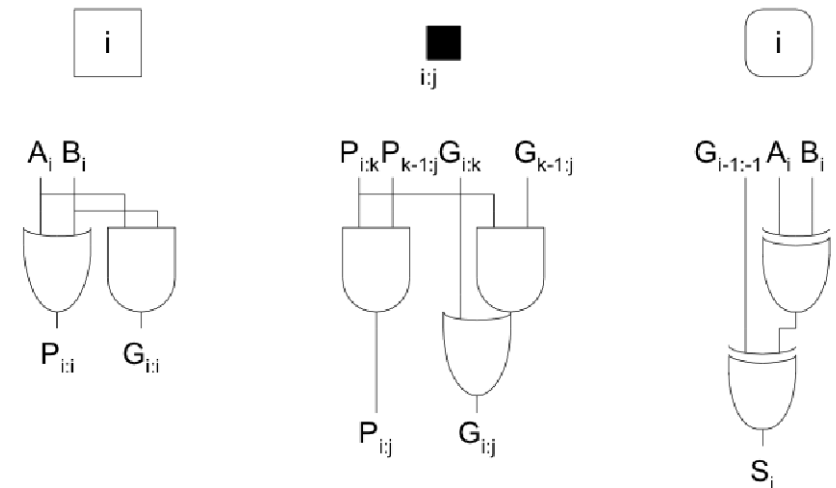


Q2: Prefix Adder

- Calculate sum of 13, 6 with a 4-bit Prefix Adder, $C_{in} = 1$
 - ✓ Compute $(G_{i:i}, P_{i:i})$ for each column
 - ✓ Compute $(G_{i:j}, P_{i:j})$ for each box
 - ✓ Compute sum S_i with $A_i, B_i, G_{i-1:-1}$ for each column



Legend



Q2: Prefix Adder

- Calculate sum of 13, 6 with a 4-bit Prefix Adder, $C_{in} = 1$
 - ✓ $1101 + 0110$
 - ✓ Compute $(G_{i:i}, P_{i:i})$ for each column

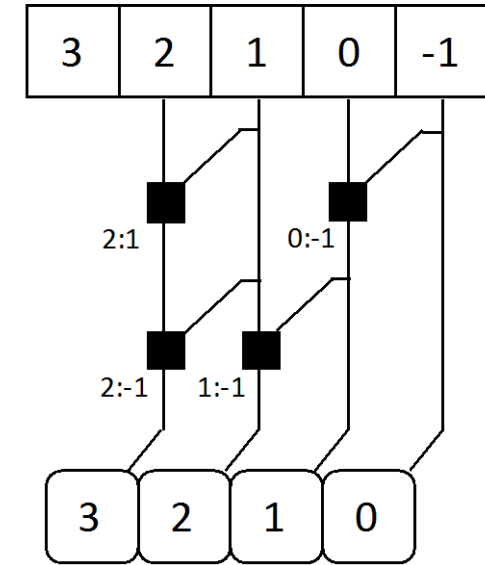
$$(G_{-1:-1}, P_{-1:-1}) = (1, 0)$$

$$(G_{0:0}, P_{0:0}) = (0, 1)$$

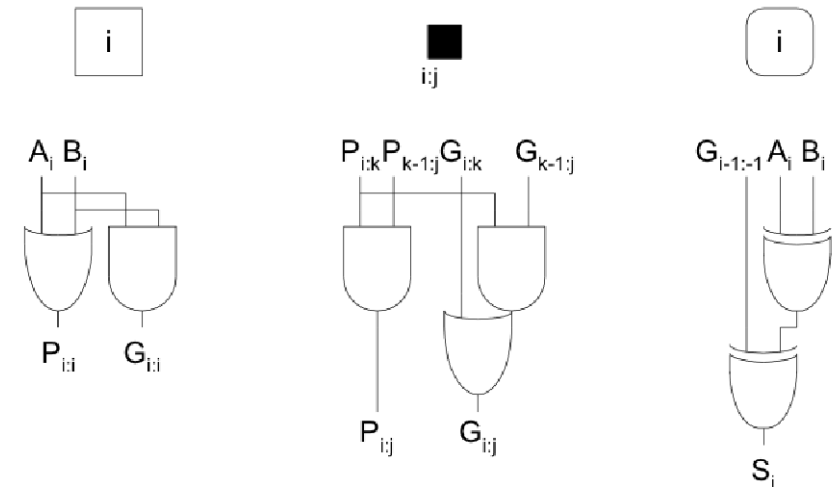
$$(G_{1:1}, P_{1:1}) = (0, 1)$$

$$(G_{2:2}, P_{2:2}) = (1, 1)$$

$$(G_{3:3}, P_{3:3}) = (0, 1)$$



Legend



Q2: Prefix Adder

- Calculate sum of 13, 6 with a 4-bit Prefix Adder, $C_{in} = 1$
 - ✓ 1101 + 0110
 - ✓ Compute $(G_{i:i}, P_{i:i})$ for each column
 - ✓ Compute $(G_{i:j}, P_{i:j})$ for each box

$$G_{i:j} = G_{i:k} + P_{i:k} G_{k-1:j}$$

$$P_{i:j} = P_{i:k} P_{k-1:j}$$

Stage 1:

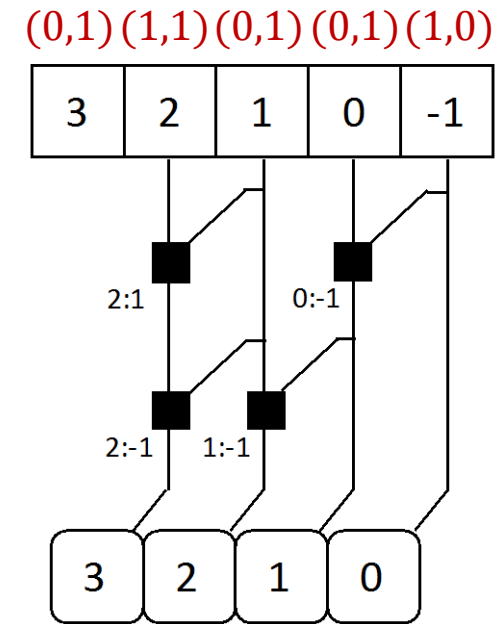
$$(G_{0:-1}, P_{0:-1}) = (1,0)$$

$$(G_{2:1}, P_{2:1}) = (1,1)$$

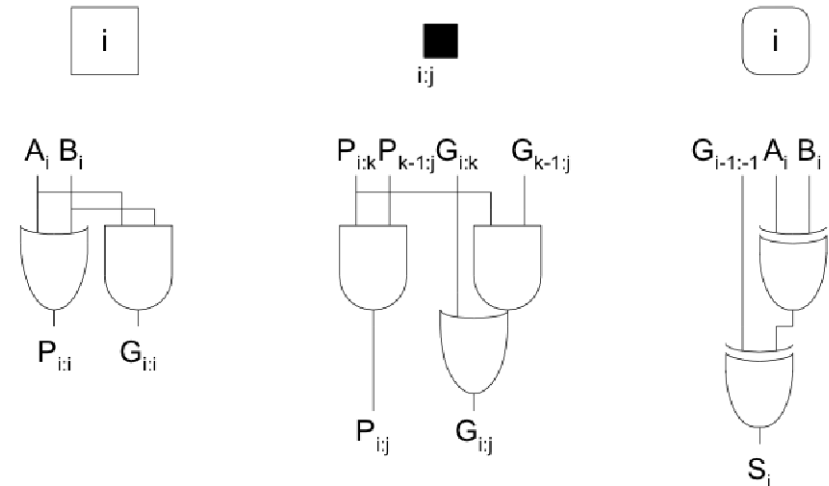
Stage 2:

$$(G_{1:-1}, P_{1:-1}) = (1,0)$$

$$(G_{2:-1}, P_{2:-1}) = (1,0)$$



Legend



Q2: Prefix Adder

- Calculate sum of 13, 6 with a 4-bit Pref Adder, $C_{in} = 1$
 - ✓ $1101 + 0110$
 - ✓ Compute $(G_{i:i}, P_{i:i})$ for each column
 - ✓ Compute $(G_{i:j}, P_{i:j})$ for each box
 - ✓ Compute sum S_i with $A_i, B_i, G_{i-1:-1}$ for each column

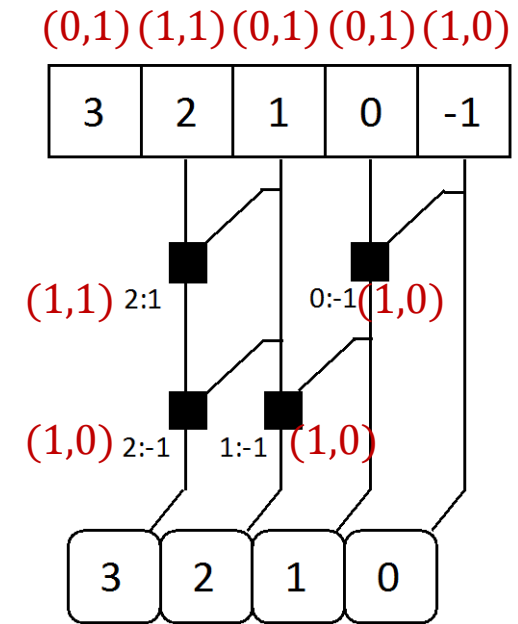
$$S_0 = (A_0 \oplus B_0) \oplus G_{-1:-1} = 0$$

$$S_1 = (A_1 \oplus B_1) \oplus G_{0:-1} = 0$$

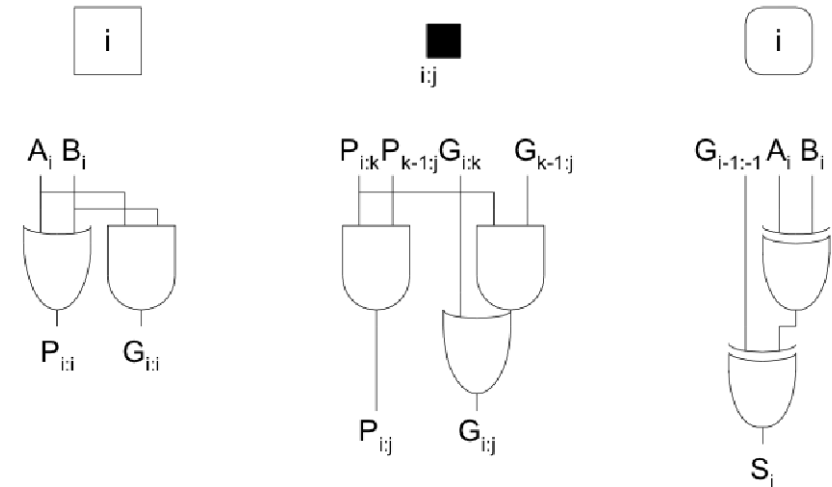
$$S_2 = (A_2 \oplus B_2) \oplus G_{1:-1} = 1$$

$$S_3 = (A_3 \oplus B_3) \oplus G_{2:-1} = 0$$

$$C_{out} = G_{3:-1} = G_{3:3} + P_{3:3}G_{2:-1} = 1$$



Legend



Q3: Carry Save Adder

- A Redundant Number System
- Save the Partial Solution for Multiple Additions
- One Bit Propagation for Each Addition

Q3: Carry Save Adder

- Example: calculate the sum of 23, 11, 17

Weight	32	16	8	4	2	1
n1=23	0	1	0	1	1	1
n2=11	0	0	1	0	1	1
n3=17	0	1	0	0	0	1

Q3: Carry Save Adder

- Example: calculate the sum of 23, 11, 17

Weight	64	32	16	8	4	2	1
n1=23	0	0	1	0	1	1	1
n2=11	0	0	0	1	0	1	1
n3=17	0	0	1	0	0	0	1

n1	00	00	10	00	10	10	10
n2	0	0	0	1	0	1	1
A=n1+n2							
n3							
B=A+n3							
Sum							

Q3: Carry Save Adder

- Example: calculate the sum of 23, 11, 17

Weight	64	32	16	8	4	2	1
n1=23	0	0	1	0	1	1	1
n2=11	0	0	0	1	0	1	1
n3=17	0	0	1	0	0	0	1

n1	00	00	10	00	10	10	10
n2	0	0	0	1	0	1	1
A=n1+n2	00	00	10	10	11	01	00
n3							
B=A+n3							
Sum							

Q3: Carry Save Adder

- Example: calculate the sum of 23, 11, 17

Weight	64	32	16	8	4	2	1
n1=23	0	0	1	0	1	1	1
n2=11	0	0	0	1	0	1	1
n3=17	0	0	1	0	0	0	1

n1	00	00	10	00	10	10	10
n2	0	0	0	1	0	1	1
A=n1+n2	00	00	10	10	11	01	00
n3	0	0	1	0	0	0	1
B=A+n3							
Sum							

Q3: Carry Save Adder

- Example: calculate the sum of 23, 11, 17

Weight	64	32	16	8	4	2	1
n1=23	0	0	1	0	1	1	1
n2=11	0	0	0	1	0	1	1
n3=17	0	0	1	0	0	0	1

n1	00	00	10	00	10	10	10
n2	0	0	0	1	0	1	1
A=n1+n2	00	00	10	10	11	01	00
n3	0	0	1	0	0	0	1
B=A+n3	00	01	00	11	00	10	10
Sum							

An extra non-redundant addition is needed!

Q3: Carry Save Adder

- Example: calculate the sum of 23, 11, 17

Weight	64	32	16	8	4	2	1
n1=23	0	0	1	0	1	1	1
n2=11	0	0	0	1	0	1	1
n3=17	0	0	1	0	0	0	1

n1	00	00	10	00	10	10	10
n2	0	0	0	1	0	1	1
A=n1+n2	00	00	10	10	11	01	00
n3	0	0	1	0	0	0	1
B=A+n3	00	01	00	11	00	10	10
Two bits of B	0	1	0	1	0	0	0
	0	0	0	1	0	1	1
Sum							

Q3: Carry Save Adder

- Example: calculate the sum of 23, 11, 17

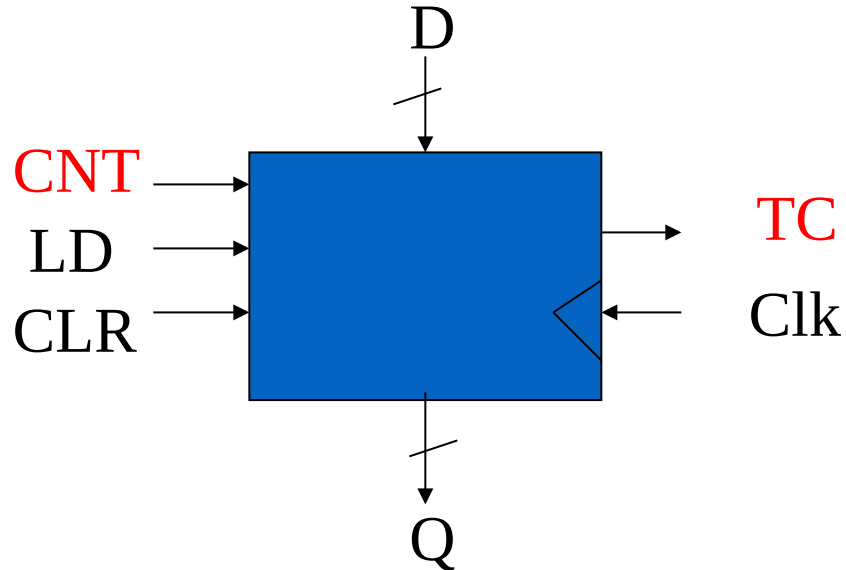
Weight	64	32	16	8	4	2	1
n1=23	0	0	1	0	1	1	1
n2=11	0	0	0	1	0	1	1
n3=17	0	0	1	0	0	0	1

n1	00	00	10	00	10	10	10
n2	0	0	0	1	0	1	1
A=n1+n2	00	00	10	10	11	01	00
n3	0	0	1	0	0	0	1
B=A+n3	00	01	00	11	00	10	10
Two bits of B	0	1	0	1	0	0	0
	0	0	0	1	0	1	1
Sum	0	1	1	0	0	1	1

Q4: Counter

- **A modulo-m counter**

- ✓ Count from 0 to m-1 when CNT = 1 and LD = 0
- ✓ Don't forget the labels



$$\begin{aligned} Q(t+1) &= (0, 0, \dots, 0) && \text{if CLR} = 1 \\ &= D && \text{if LD} = 1 \text{ and CLR} = 0 \\ &= (Q(t) + 1) \bmod n && \text{if LD} = 0, \text{CNT} = 1 \text{ and CLR} = 0 \\ &= Q(t) && \text{if LD} = 0, \text{CNT} = 0 \text{ and CLR} = 0 \\ \text{TC} &= 1 && \text{if } Q(t) = n-1 \text{ and CNT} = 1 \\ &= 0 && \text{otherwise} \end{aligned}$$

Q4: Counter

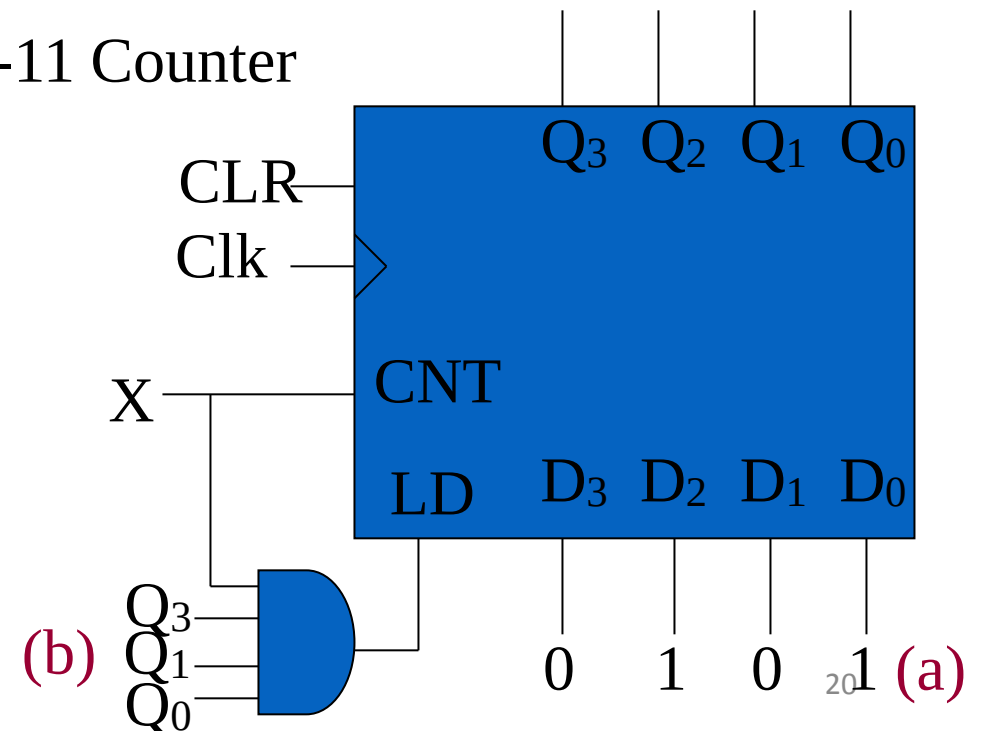
- **Design a a-to-b counter with modulo-16 counter**

- ✓ Count from a to b ($0 < a < b < 15$), a 5-to-11 counter from slide 14 of Lecture 15
- ✓ Set $(D_3D_2D_1D_0) = a$
- ✓ When $CNT = 1$ and $(Q_3Q_2Q_1Q_0) = b$, set **LD = 1** then $D \rightarrow Q(t + 1)$
- ✓ Comb logic for LD only takes 1s

$$(Q_3Q_2Q_1Q_0) = 1011$$

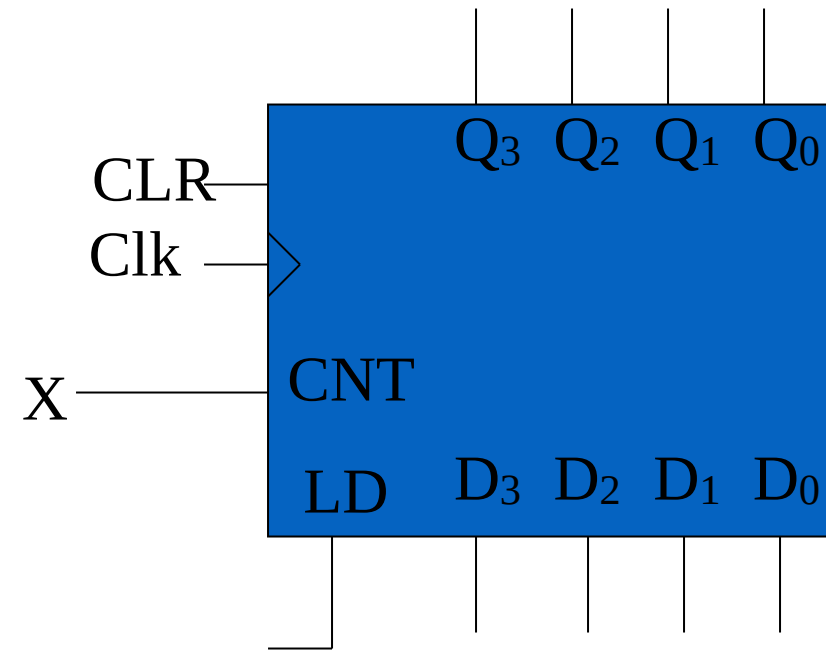
$(Q_3Q_2Q_1Q_0) = \mathbf{1111}$ cannot be reached in
this 5-to-11 counter!

A 5-to-11 Counter



Q4: Counter

- **Design a counter of an arbitrary sequence with a modulo-16 counter**
 - ✓ an arbitrary sequence 3,4,5, 0,1,2, 8,9
 - ✓ when $Q = 5$, load $D = 0$
 - ✓ when $Q = 2$, load $D = 8$
 - ✓ when $Q = 9$, load $D = 3$



Q4: Counter

- **Design a counter of an arbitrary sequence with a modulo-16 counter**

- ✓ an arbitrary sequence 3,4,5, 0,1,2, 8,9

- when $Q = 5$, load $D = 0$

- when $Q = 2$, load $D = 8$

- when $Q = 9$, load $D = 3$

- ✓ Set LD = 1 for load new number, otherwise LD = 0. If number not in the sequence, set LD as don't care.

- ✓ Set D = new number when LD = 1, otherwise set D as don't cares.

Id	$Q_3Q_2Q_1Q_0$	LD	D_3	D_2	D_1	D_0
0	0000	0	-	-	-	-
1	0001	0	-	-	-	-
2	0010	1	1	0	0	0
3	0011	0	-	-	-	-
4	0100	0	-	-	-	-
5	0101	1	0	0	0	0
8	1000	0	-	-	-	-
9	1001	1	0	0	1	1
6,7,10~15	0110, 0111, 1010~1111	-	-	-	-	-

Q4: Counter

- Design a counter of an arbitrary sequence with a modulo-16 counter

- ✓ an arbitrary sequence 3,4,5, 0,1,2, 8,9

- ✓ Use K-maps

$$LD = Q_1Q'_0 + Q_2Q_0 + Q_3Q_0$$

$Q_1Q_0 \backslash Q_3Q_2$	00	01	11	10
00	0 ⁰	0 ⁴	X ¹²	0 ⁸
01	0 ¹	1 ⁵	X ¹³	1 ⁹
11	0 ³	X ⁷	X ¹⁵	X ¹¹
10	1 ²	X ⁶	X ¹⁴	X ¹⁰

$D_3 = ?$

$D_2 = ?$

$D_1 = ?$

$D_0 = ?$

Id	$Q_3Q_2Q_1Q_0$	LD	D_3	D_2	D_1	D_0
0	0000	0	-	-	-	-
1	0001	0	-	-	-	-
2	0010	1	1	0	0	0
3	0011	0	-	-	-	-
4	0100	0	-	-	-	-
5	0101	1	0	0	0	0
8	1000	0	-	-	-	-
9	1001	1	0	0	1	1
6,7,10~15	0110, 0111, 1010~1111	-	-	-	-	-