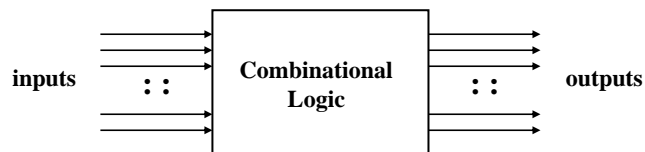


## فصل چهارم : مدارهای ترکیبی Combinational Circuits

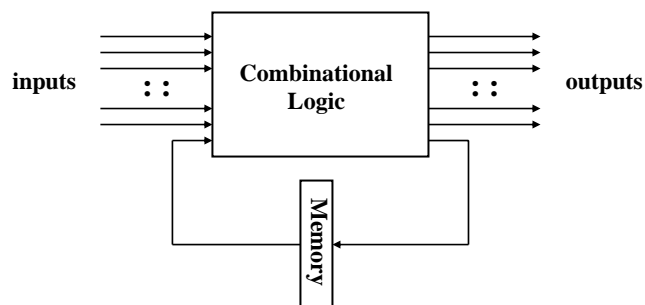
- دودسته مدارهای منطقی :
  - ❖ ترکیبی یا combinational
  - ❖ ترتیبی یا sequential
- مدارهای ترکیبی :



خروجی های هر لحظه فقط به ورودی های همان لحظه بستگی دارد.

## مدارهای ترکیبی Combinational Circuits

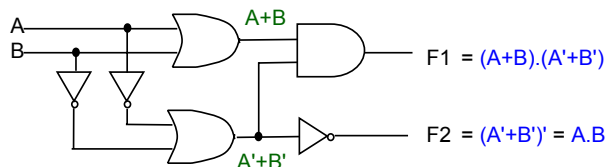
- مدارهای ترتیبی



خروجی ها به ورودی های فعلی و قبلی (وضعیت قبلی مدار) بستگی دارند  
عناصر حافظه (از طریق فیدبک) حاوی اطلاعات قبلی هستند.

## روش تحلیل Analysis Procedure

با داشتن یک مدار ترکیبی آیا می توان تابع مربوطه را تحلیل کرد؟



### Steps:

- ❖ 1. Label the inputs and outputs.
- ❖ 2. Obtain the functions of intermediate points and the outputs.
- ❖ 3. Draw the truth table.
- ❖ 4. Deduce the functionality of the circuit ➔ half adder.

A	B	(A+B)	(A'+B)'	F1	F2
0	0	0	1	0	0
0	1	1	1	1	0
1	0	1	1	1	0
1	1	1	0	0	1

## روشهای طراحی Design Methods

- روشهای مختلف طراحی مدار های ترکیبی :
  - ❖ Gate-level method (with logic gates)
  - ❖ Block-level design method
- تقسیم بندی IC ها بر اساس تعداد گیت های داخلی:
  - ❖ Small-scale integration (SSI): up to 12 gates
  - ❖ Medium-scale integration (MSI): 12-99 gates
  - ❖ Large-scale integration (LSI): 100-9999 gates
  - ❖ Very large-scale integration (VLSI): 10,000-99,999 gates
  - ❖ Ultra large-scale integration (ULSI): > 100,000 gates
- اهداف اصلی در طراحی مدار:
  - ❖ (i) کاهش هزینه
    - کاهش تعداد گیتها
    - کاهش تعداد IC ها
  - ❖ (ii) افزایش سرعت
  - ❖ (iii) سادگی طرح

## Gate-level (SSI) Design: Half Adder

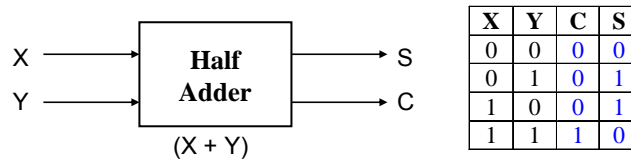
▪ روش طراحی

(1) بیان مسئله

Example: Build a **Half Adder** to add two bits

(2) خروجی ها و ورودی ها را مشخص و اسم گذاری کنید:

Example: Two inputs and two outputs labelled, as follows:



(3) جدول درستی را رسم کنید.

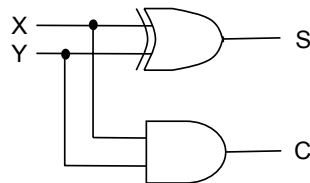
## Gate-level (SSI) Design: Half Adder

(4) عبارت بولی ساده شده را برای خروجی ها بدست آورید:

Example:  $C = XY$   
 $S = X'Y + XY' = X \oplus Y$

X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

(5) دیاگرام منطقی را رسم کنید:



Half Adder

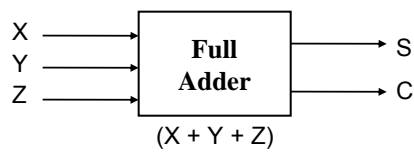
## Gate-level (SSI) Design: Full Adder

- Half-adder فقط دو بیت را با هم جمع می کند:
- برای جمع دو عدد باینری نیاز به جمع سه بیت (شامل رقم نقلی) داریم.

▪ Example:

$$\begin{array}{r}
 1\ 1\ 1\ \text{carry} \\
 0\ 0\ 1\ 1\ X \\
 +\ 0\ 1\ 1\ 1\ Y \\
 \hline
 1\ 0\ 1\ 0\ S
 \end{array}$$

- برای این منظور نیازمند Full Adder خواهیم بود.



## Gate-level (SSI) Design: Full Adder

▪ Truth table:

X	Y	Z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

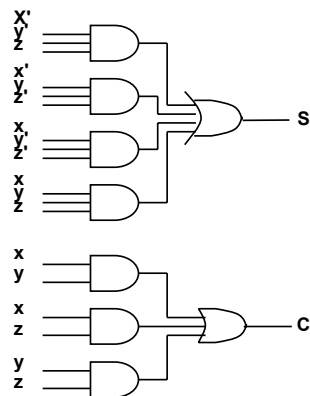
		C			
		00	01	11	10
x	0			1	
	1		1	1	1

		S			
		00	01	11	10
x	0		1		1
	1	1		1	

Note:

Z - carry in (to the current position)

C - carry out (to the next position)



▪ Using K-map, simplified SOP form:

$$C = XY + XZ + YZ$$

$$S = X'Y'Z + X'YZ' + XY'Z' + XYZ$$

## Gate-level (SSI) Design: Full Adder

- Alternative formulae using algebraic manipulation:

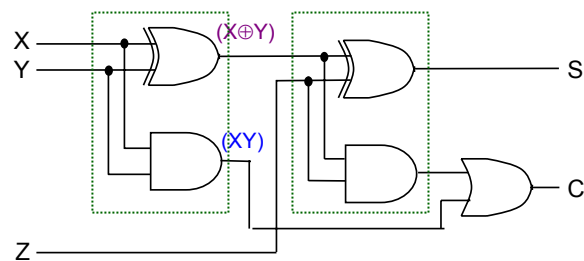
$$\begin{aligned}C &= XY + XZ + YZ \\ &= XY + (X + Y)Z \\ &= XY + ((X \oplus Y) + XY)Z \\ &= XY + (X \oplus Y)Z + XYZ \\ &= XY + (X \oplus Y)Z\end{aligned}$$

$$\begin{aligned}S &= X'Y'Z + X'YZ' + XY'Z' + XYZ \\ &= X'(Y'Z + YZ') + X(Y'Z' + YZ) \\ &= X'(Y \oplus Z) + X(Y \oplus Z)' \\ &= X \oplus (Y \oplus Z) \text{ or } (X \oplus Y) \oplus Z\end{aligned}$$

## Gate-level (SSI) Design: Full Adder

$$C = XY + (X \oplus Y)Z$$

$$S = (X \oplus Y) \oplus Z$$

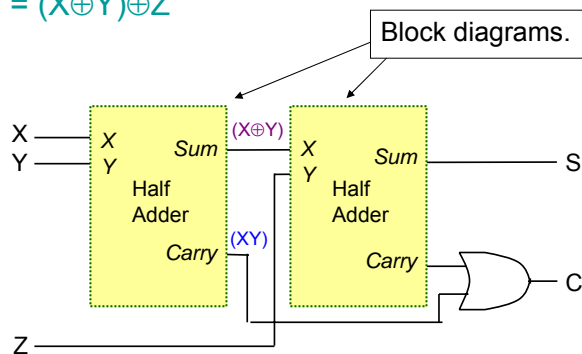


Full Adder made from two Half-Adders (+ OR gate).

## Gate-level (SSI) Design: Full Adder

$$C = XY + (X \oplus Y)Z$$

$$S = (X \oplus Y) \oplus Z$$

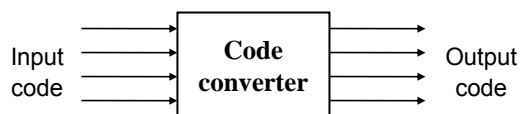


Full Adder made from two Half-Adders (+ OR gate).

## مبدل های کد

### Code Converters

- مبدل کد مداری است که یک نوع کد را به کد دیگر ترجمه می کند.



- Example: **BCD to Excess-3 Code Converter.**

*Input:* BCD digit

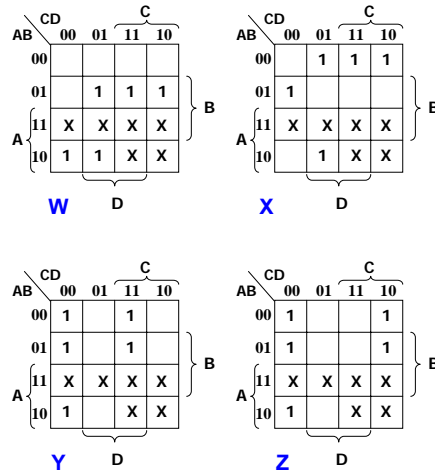
*Output:* Excess-3 digit

# BCD-to-Excess-3 Code Converter

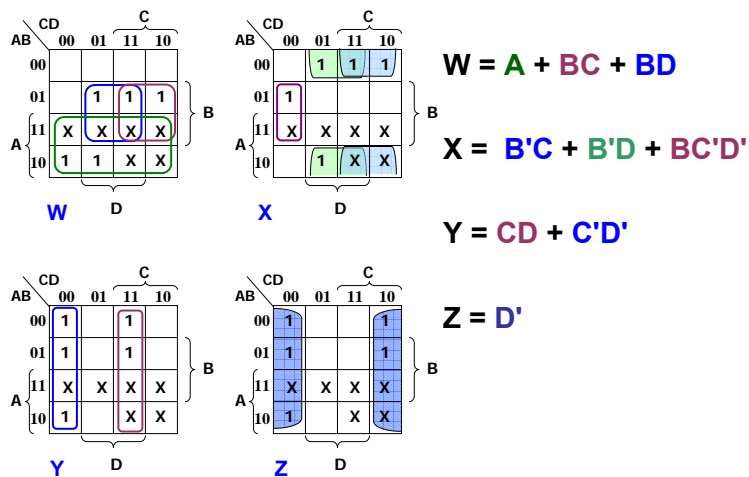
Truth table:

	BCD				Excess-3			
	A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0
10	1	0	1	0	X	X	X	X
11	1	0	1	1	X	X	X	X
12	1	1	0	0	X	X	X	X
13	1	1	0	1	X	X	X	X
14	1	1	1	0	X	X	X	X
15	1	1	1	1	X	X	X	X

K-maps:



# BCD-to-Excess-3 Code Converter

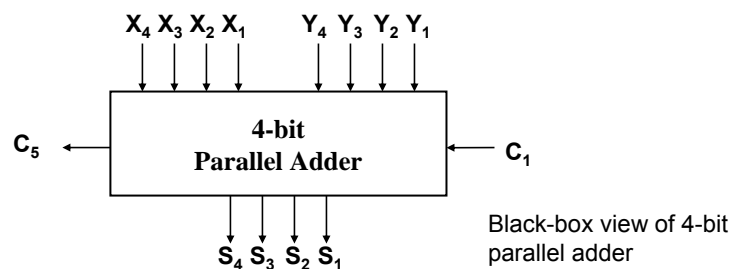


## Block-Level Design Method

- مدارات پیچیده تر می تواند با روش block-level طراحی شود.
  - در این روش مسئله اصلی به قسمت های کوچکتر شکسته شده و طوری تجزیه می گردد که با بلوک های کوچکتر قابل پیاده سازی باشد.
  - مثال : استفاده از بلوک های جمع کننده موازی ۴ بیتی برای طراحی :
- ❖ (1) BCD-to-Excess-3 Code Conversion
  - ❖ (2) 16-Bit Parallel Adder
  - ❖ (3) Adder cum Subtractor

## جمع کننده موازی ۴ بیتی 4-bit Parallel Adder

- مداری را در نظر بگیرید که دو عدد ۴ بیتی و یک رقم نقلی ورودی را با هم جمع کرده و نتیجه ۵ رقمی را تولید می کند.



- ۵ بیت خروجی کافیت چون بزرگترین نتیجه عبارتست از:

$$(1111)_2 + (1111)_2 + (1)_2 = (11111)_2$$

## جمع کننده موازی ۴ بیتی 4-bit Parallel Adder

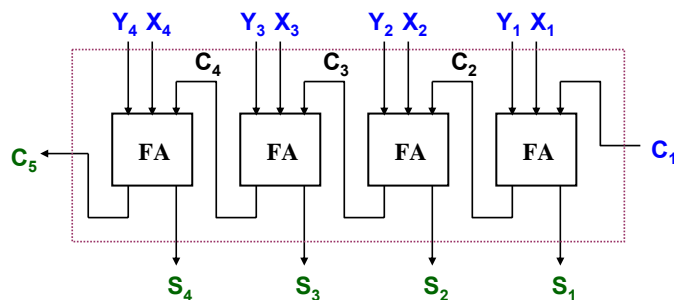
- روش طراحی SSI (Gate-level) نمی تواند استفاده گردد.
- جدول ۹ ورودی بسیار بزرگ خواهد بود و دارای  $2^9=512$  سطر می باشد.

$X_4X_3X_2X_1$	$Y_4Y_3Y_2Y_1$	$C_1$	$C_5$	$S_4S_3S_2S_1$
0 0 0 0	0 0 0 0	0	0	0 0 0 0
0 0 0 0	0 0 0 0	1	0	0 0 0 1
0 0 0 0	0 0 0 1	0	0	0 0 0 1
...	...	...	...	...
0 1 0 1	1 1 0 1	1	1	0 0 1 1
...	...	...	...	...
1 1 1 1	1 1 1 1	1	1	1 1 1 1

- ساده سازی نیز بسیار مشکل خواهد بود.

## جمع کننده موازی ۴ بیتی 4-bit Parallel Adder

- با در نظر گرفتن ۴ تمام جمع کننده بصورت پشت سر هم (Cascading) خواهیم داشت:



- این مدار **Parallel Adder** نامیده می شود چرا که تمام ورودی ها بطور موازی و همزمان به مدار اعمال می شود. همچنین **Ripple-Carry Adder** نیز نامیده می شود.

## BCD-to-Excess-3 Code Converter

	BCD				Excess-3			
	A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0
10	1	0	1	0	X	X	X	X
11	1	0	1	1	X	X	X	X
12	1	1	0	0	X	X	X	X
13	1	1	0	1	X	X	X	X
14	1	1	1	0	X	X	X	X
15	1	1	1	1	X	X	X	X

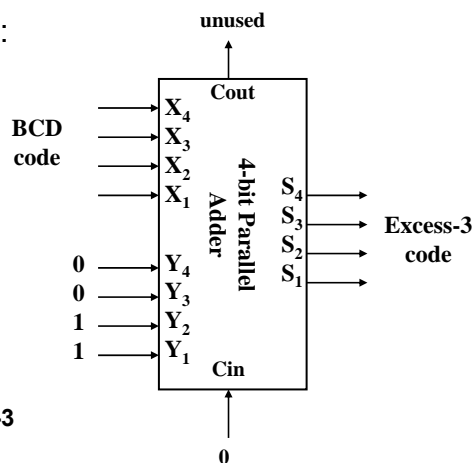
- روش طراحی **Gate-level** می تواند استفاده گردد. چون فقط ۴ ورودی داریم.
- با این وجود روش دیگری نیز وجود دارد.
- با در نظر گرفتن رابطه زیر :

**Excess-3 Code**  
 $= \text{BCD Code} + (0011)_2$

## BCD-to-Excess-3 Code Converter

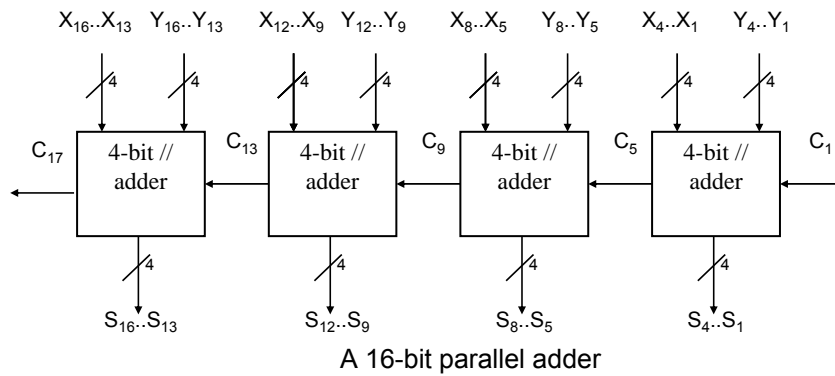
$\text{Excess-3 Code} = \text{BCD Code} + (0011)_2$

- Block-level circuit:



## 16-bit Parallel Adder

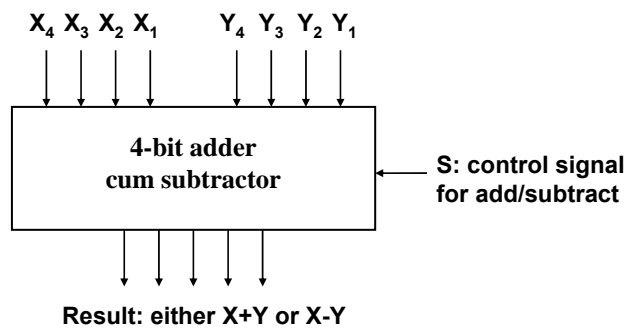
- جمع کننده های موازی بزرگتر را می توان از انواع بزرگتر آن بدست آورد.
- Example: a 16-bit parallel adder can be constructed from four 4-bit parallel adders:



## جمع کننده-تفریق گر موازی ۴ بیتی

### 4-bit Parallel Adder cum Sub tractor

- با روش مکمل ۲ می توان از جمع کننده بعنوان تفریق گر استفاده کرد.
- بنا بر این میتوان با استفاده از جمع کننده موازی مدار طراحی کرد که هم جمع و هم تفریق انجام دهد.



## جمع کننده-تفریق گر موازی ۴ بیتی

▪ سیگنال کنترل:

**S=0** به معنی جمع

**S=1** به معنی تفریق

▪ یادآوری می کنیم که:

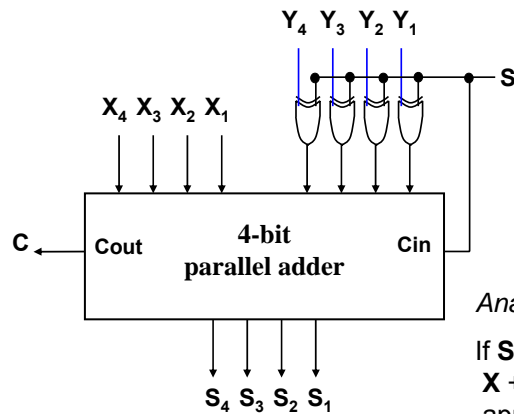
$$\begin{aligned} X-Y &= X + (-Y) \\ &= X + (2\text{'s complement of } Y) \\ &= X + (1\text{'s complement of } Y) + 1 \\ X+Y &= X + \underline{Y} \end{aligned}$$



output = Y when S=0  
= Y' when S=1

## 4-bit Parallel Adder cum Subtractor

▪ Adder cum subtractor circuit:



A 4-bit adder cum subtractor

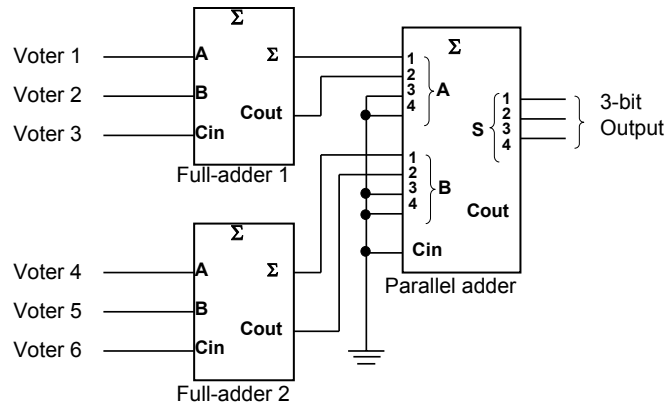
*Analysis:*

If **S=1**, then  
**X + (1's complement of Y) + 1**  
appears as the result.

If **S=0**, then **X+Y** appears as  
the result.

## Arithmetic Circuits: Cascading Adders

- کاربرد: سیستم رای گیری ۶ نفره (6-person voting system)
  - ❖ Use FAs and a 4-bit binary parallel adder.
  - ❖ Each FA can sum up to 3 votes.



## Arithmetic Circuits: Comparator

- Magnitude comparator دو عدد  $A$  و  $B$  را با هم مقایسه و نتایج زیر را مشخص می کند:  $A > B$ ,  $A = B$  or  $A < B$

- روش کلاسیک نیاز به  $2^{2n}$  سطر در جدول درستی دارد.

- از نحوه مقایسه دو عدد ۴ بیتی و  $A(a_3a_2a_1a_0)$   $B(b_3b_2b_1b_0)$  سود می بریم:

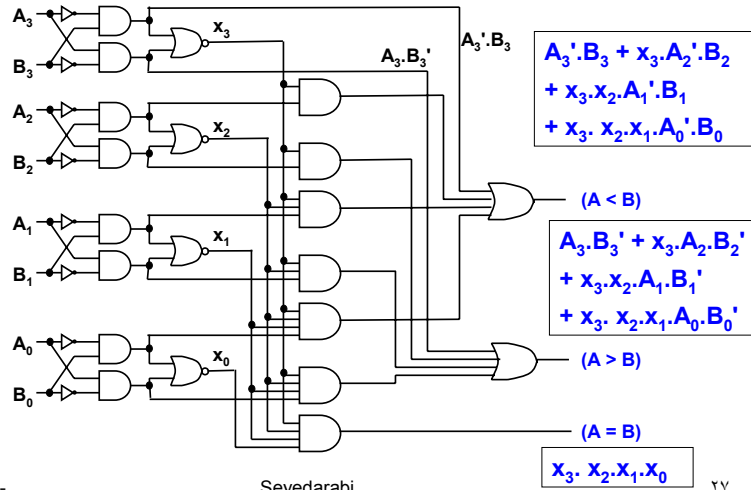
If  $(a_3 > b_3)$  then  $A > B$

If  $(a_3 < b_3)$  then  $A < B$

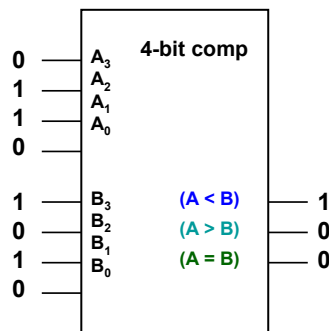
If  $(a_3 = b_3)$  then if  $(a_2 > b_2)$  ....

# Arithmetic Circuits: Comparator

Let  $A = A_3A_2A_1A_0$ ,  $B = B_3B_2B_1B_0$ ;  $x_i = A_i \cdot B_i + A_i' \cdot B_i'$



# Arithmetic Circuits: Comparator



Block diagram of a 4-bit magnitude comparator