Engr354: Digital Logic Circuits

Chapter 5: Number Representation and Arithmetic Circuits

Curtis Nelson

Number Systems and Their Representations

In this chapter you will learn about:

- Representation of numbers in computers;
- Circuits used to perform arithmetic operations;
- Performance issues in large circuits.

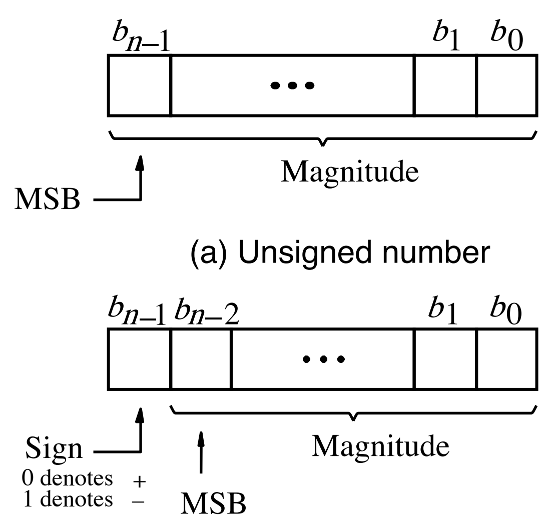
7-Bit ASCII Code

Bit positions 3210	Bit positions 654: 000	001	010	011	100	101	110	111
0000	NUL	DLE	SPACE	0	@	P	´	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	”	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	’	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	1	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	—	o	DEL

NUL	Null/Idle	SI	Shift in
SOH	Start of header	DLE	Data link escape
STX	Start of text	DC1-DC4	Device control
ETX	End of text	NAK	Negative acknowledgement
EOT	End of transmission	SYN	Synchronous idle
ENQ	Enquiry	ETB	End of transmitted block
ACQ	Acknowledgement	CAN	Cancel (error in data)
BEL	Audible signal	EM	End of medium
BS	Back space	SUB	Special sequence
HT	Horizontal tab	ESC	Escape
LF	Line feed	FS	File separator
VT	Vertical tab	GS	Group separator
FF	Form feed	RS	Record separator
CR	Carriage return	US	Unit separator
SO	Shift out	DEL	Delete/Idle

Bit positions of code format = 6 5 4 3 2 1 0

Unsigned Vs. Signed Numbers



(b) Signed number

Interpretation of Four-Bit Signed Integers

abcd	Sign and magnitude	1's complement	2's complement
0111	+7	+7	+7
0110	+6	+6	+6
0101	+5	+5	+5
0100	+4	+4	+4
0011	+3	+3	+3
0010	+2	+2	+2
0001	+1	+1	+1
0000	+0	+0	0
1000	-0	-7	-8
1001	-1	-6	-7
1010	-2	-5	-6
1011	-3	-4	-5
1100	-4	-3	-4
1101	-5	-2	-3
1110	-6	-1	-2
1111	-7	-0	-1

Sign and Magnitude

- Magnitude of positive and negative numbers represented in the same way;
- Sign used to distinguish them;
- Simple to understand;
- Complicates hardware design.

Sign and Magnitude: Add/Subtract

- If both operands have the same sign, then addition is simple
  - Add magnitudes and copy the sign.
- If they have opposite signs, then must subtract smaller from the larger.
- This is complicated, so sign and magnitude is not used in computers anymore.

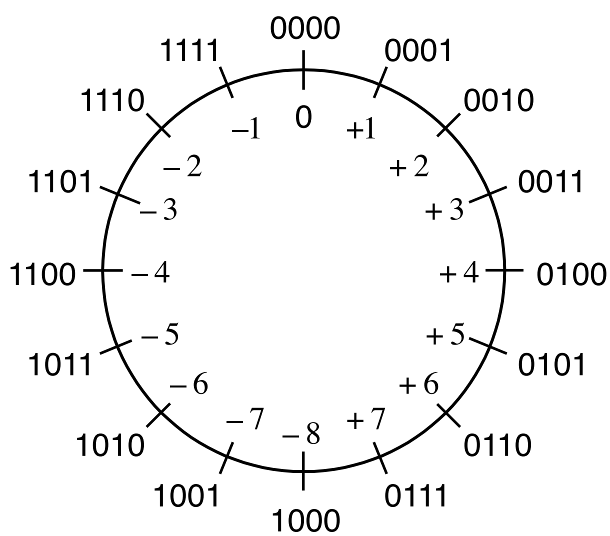
1's Complement

- n-bit negative number found by subtracting its positive form from 2^n-1
  - $K_1 = (2^n - 1) - P$
- Found by just complementing each bit.
- Used in earlier computers (pre-1970) but made hardware difficult with two representations for 0.

2's Complement

- n-bit negative number found by subtracting its positive form from 2^n
  - $K_2 = 2^n - P$
  - $K_2 = K_1 + 1$
- Complement each bit and add 1.
- Probably the only form for signed numbers used in computers today.

Graphical Interpretation of Four-bit 2's Complement Numbers



2’s Complement Addition Examples

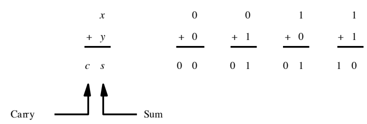
(+5)	0 1 0 1	(–5)	1 0 1 1
+ (+2)	+ 0 0 1 0	+ (+2)	+ 0 0 1 0
(+7)	0 1 1 1	(–3)	1 1 0 1

(+5)	0 1 0 1	(–5)	1 0 1 1
+ (–2)	+ 1 1 1 0	+ (–2)	+ 1 1 1 0
(+3)	1 0 0 1 1	(–7)	1 1 0 0 1
	↑ ignore		↑ ignore

2’s Complement Subtraction Examples

(+5)	0 1 0 1		0 1 0 1
– (+2)	– 0 0 1 0	⟹	+ 1 1 1 0
(+3)			1 0 0 1 1
			↑ ignore
(–5)	1 0 1 1		1 0 1 1
– (+2)	– 0 0 1 0	⟹	+ 1 1 1 0
(–7)			1 1 0 0 1
			↑ ignore
(+5)	0 1 0 1		0 1 0 1
– (–2)	– 1 1 1 0	⟹	+ 0 0 1 0
(+7)			0 1 1 1
(–5)	1 0 1 1		1 0 1 1
– (–2)	– 1 1 1 0	⟹	+ 0 0 1 0
(–3)			1 1 0 1

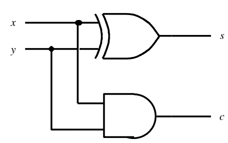
Adder Circuits – Half Adder



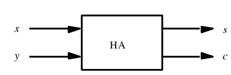
(a) The four possible cases

x	y	Carry c	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

(b) Truth table



(c) Circuit



(d) Graphical symbol

Adder Circuits – Full Adder

c_i	x_i	y_i	c_{i+1}	s_i
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

(a) Truth table

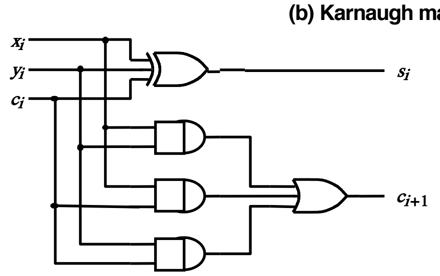
c_i \ x_iy_i	00	01	11	10
0		1		1
1	1		1	

$s_i = x_i \oplus y_i \oplus c_i$

c_i \ x_iy_i	00	01	11	10
0			1	
1		1	1	1

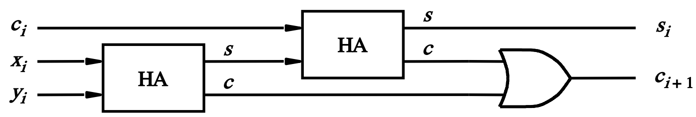
$c_{i+1} = x_iy_i + x_ic_i + y_ic_i$

(b) Karnaugh maps

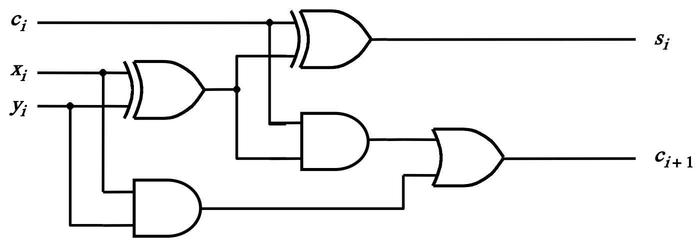


(c) Circuit

Alternative Full Adder

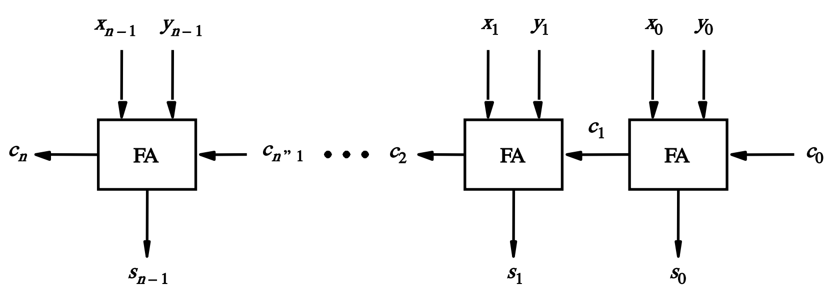


(a) Block diagram



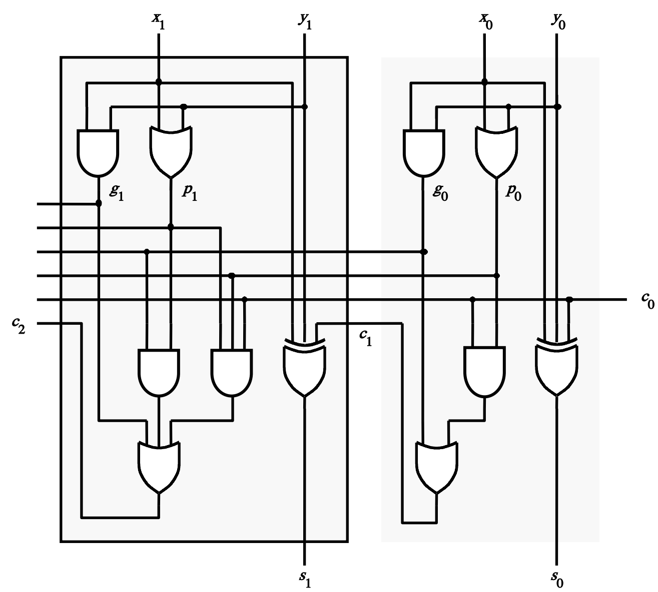
(b) Detailed diagram

N-bit Ripple Carry Adder

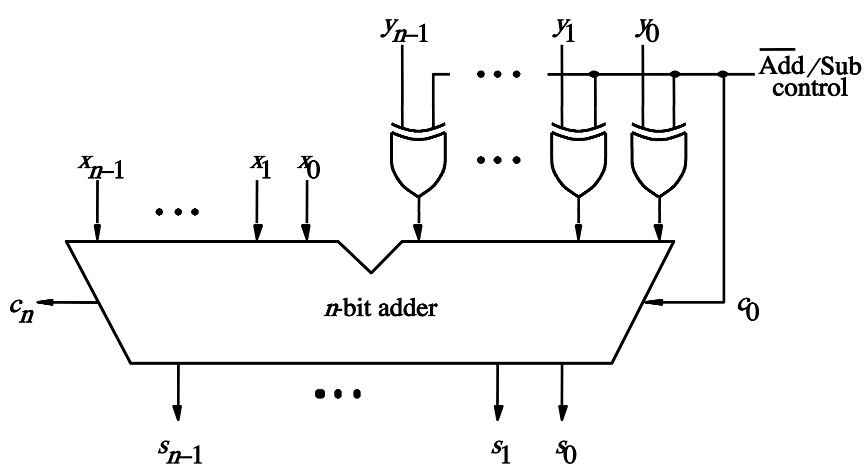


- Major issue – propagation delay.

Faster Adders – Carry Lookahead



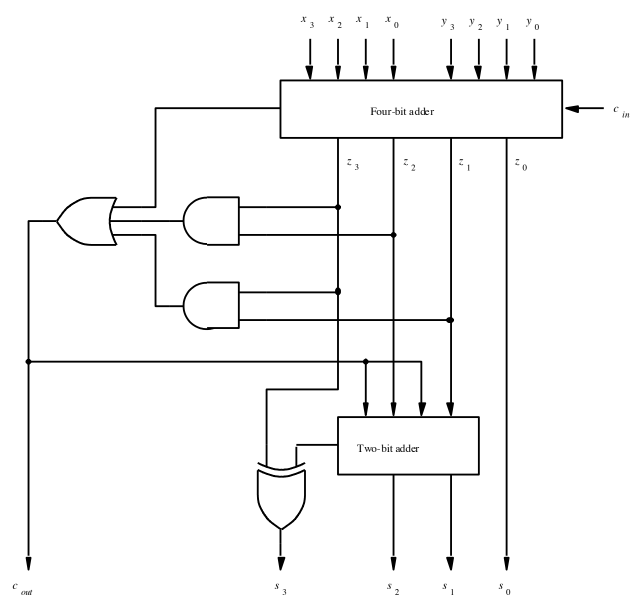
Adder / Subtractor Circuit



Binary Coded Decimal (BCD) Digits

Decimal digit	BCD code
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

One Digit BCD Adder



Other Arithmetic Circuits

- Multipliers;
- Dividers;
- Floating-point Processors.

Summary

In this chapter you learned about:

- Representation of numbers in computers;
- Circuits used to perform arithmetic operations;
- Performance issues in large circuits.