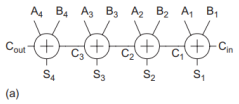
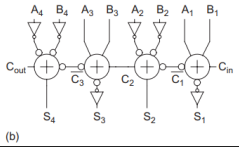
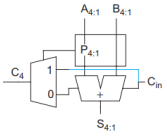
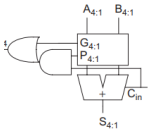
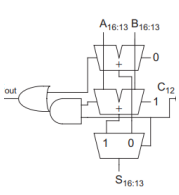
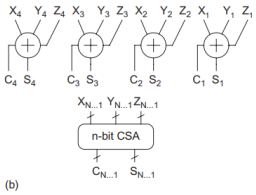


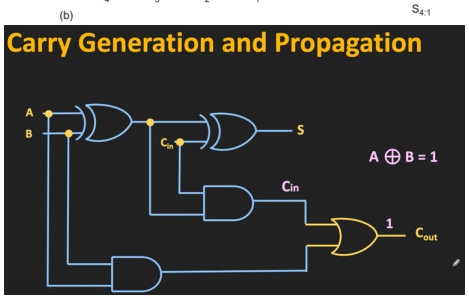




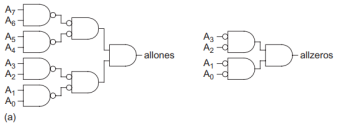


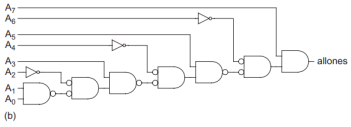


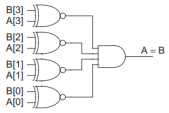




Relation	Unsigned Comparison	Signed Comparison
$A = B$	Z	Z
$A \neq B$	$\overline{Z}$	$\overline{Z}$
$A < B$	$C \cdot \overline{Z}$	$\overline{S} \cdot \overline{Z}$
$A > B$	$\overline{C}$	S
$A \leq B$	C	$\overline{S}$
$A \geq B$	$\overline{C} + Z$	$S + Z$







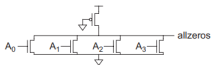


FIGURE 11.46 Equality comparator

- Logical Shift:
  - Shifts number left or right and fills with 0's
    - 1011 LSR 1 = 0101 1011 LSL1 = 0110
- Arithmetic Shift:
  - Shifts number left or right. Rt shift sign extends
    - 1011 ASR1 = 1101 1011 ASL1 = 0110
    - ASR of negative number remains negative (divide by power of 2)
  - ASL = LSL *(both stuff 0s in from the right/LSB side)*
- Rotate:
  - Shifts number left or right and fills with lost bits
    - 1011 ROR1 = 1101 1011 ROL1 = 0111

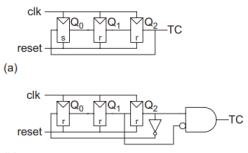


FIGURE 11.53 3-bit ring and Johnson counters

TABLE 11.6 Johnson counter sequence

Cycle	Q_0	Q_1	Q_2	TC
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	0	1	1	0
5	0	0	1	1
6	0	0	0	0
Repeats forever				

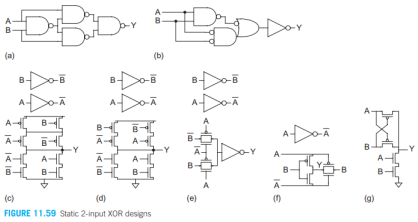
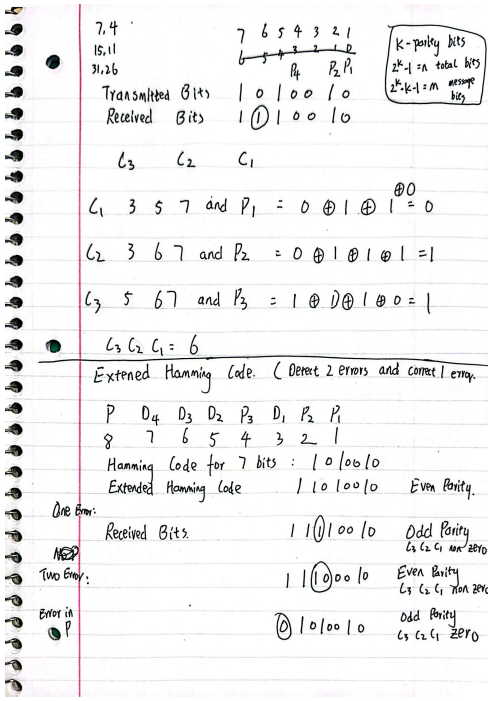


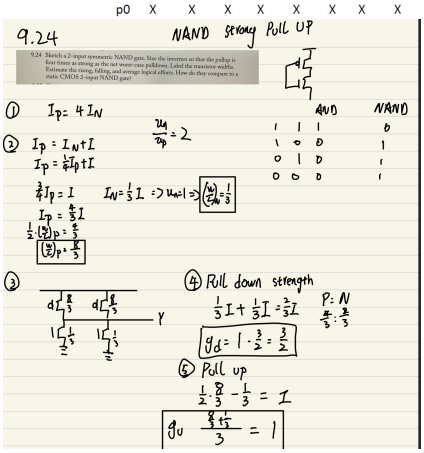
FIGURE 11.59 Static 2-input XOR designs

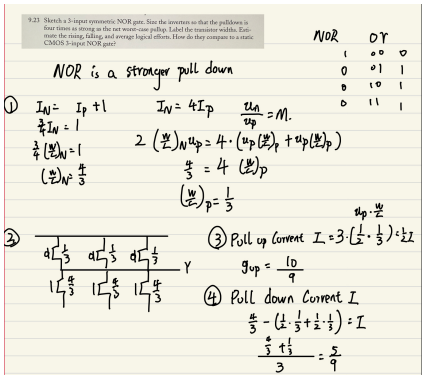
X marks the spot -- positions of XORs

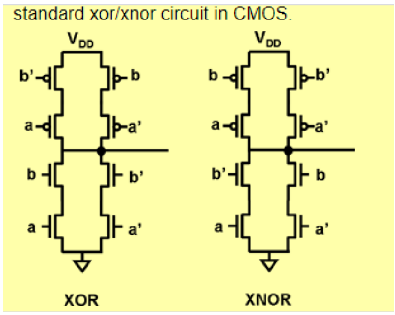
p1, p2, p4, and p8 provide SEC; global parity p0 extends this to DED
Note positions and names of parity bits (always powers of 2 -- we'll see why)

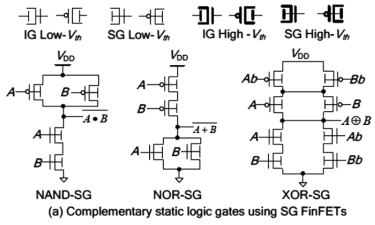
Bit position		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Encoded data bits		p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8	d9	d10	d11
Parity bit coverage	p1	X		X		X		X		X		X		X		X
	p2		X	X			X	X			X	X			X	X
	p4				X	X	X	X					X	X	X	X
	p8								X	X	X	X	X	X	X	X
	p0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X



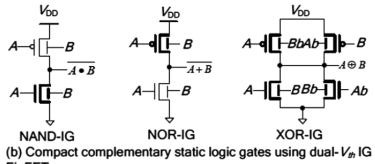




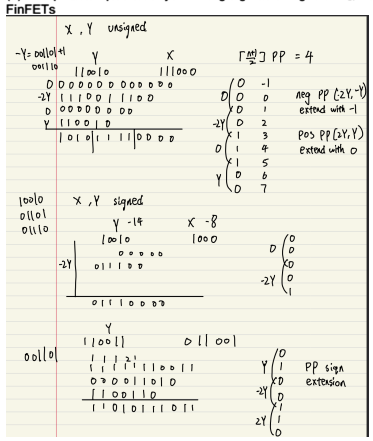


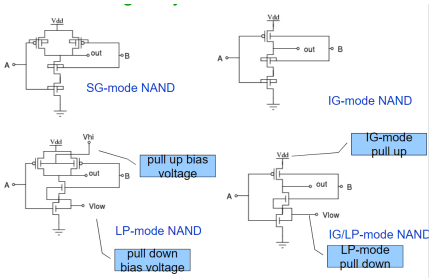


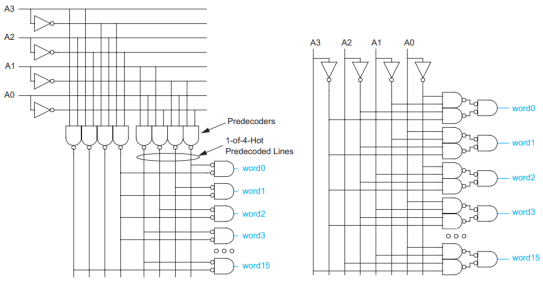
(a) Complementary static logic gates using SG FinFETs



(b) Compact complementary static logic gates using dual-V_{th} IG FinFETs







Gate type	Logical effort	Formula	$n=2$ $\gamma=2$	$n=3$ $\gamma=2$	$n=4$ $\gamma=2$
NAND	total	$\frac{n(n+\gamma)}{1+\gamma}$	8/3	5	8
	per input	$\frac{(n+\gamma)}{1+\gamma}$	4/3	5/3	2
NOR	total	$\frac{n(1+n\gamma)}{1+\gamma}$	10/3	7	12
	per input	$\frac{(1+n\gamma)}{1+\gamma}$	5/3	7/3	3
multiplexer	total	$4n$	8	12	16
	d, $s*$	2,2	2,2	2,2	2,2
XOR, XNOR, parity (symmetric)	total	$n^2 2^{n-1}$	8	36	128
	per bundle	$n2^{n-1}$	4	12	32
XOR, XNOR, parity (asymmetric)	total per bundle		8 4,4	24 6,12,6	48 8,16,16,8
majority (symmetric)	total			12	
	per input			4	
majority (asymmetric)	total			10	
	per input			4,4,2	
C-element	total	n^2	4	9	16
	per input	n	2	3	4
latch (dynamic)	total	4			
	d, $\phi*$	2,2			
upper bounds	total	$\frac{\gamma n^2+n}{1+\gamma}$	32/3	48	512/3
	per bundle	$\frac{\gamma n2^n}{1+\gamma}$	16/3	16	128/3

Table 4.1: Summary of calculations of the logical effort of logic gates.

Gate type	Formula	Parasitic delay when $p_{inv}=1.0$ $n=1$	$n=2$	$n=3$	$n=4$
inverter	p_{inv}	1			
NAND	np_{inv}		2	3	4
NOR	np_{inv}		2	3	4
multiplexer	$2np_{inv}$		4	6	8
XOR, XNOR, parity	$n2^{n-1}p_{inv}$		4	12	
majority	$6p_{inv}$			6	
C-element	np_{inv}		2	3	4
latch	$2p_{inv}$	2			

Table 4.2: Estimates of the parasitic delay of logic gates.

TABLE 11.12 Radix-4 modified Booth encoding values

Inputs			Partial Product	Booth Selects		
x_{2i+1}	x_{2i}	x_{2i-1}	PP_i	SINGLE$_i$	DOUBLE$_i$	NEG$_i$
0	0	0	0	0	0	0
0	0	1	Y	1	0	0
0	1	0	Y	1	0	0
0	1	1	$2Y$	0	1	0
1	0	0	$-2Y$	0	1	1
1	0	1	$-Y$	1	0	1
1	1	0	$-Y$	1	0	1
1	1	1	$-0\ (=0)$	0	0	1

Example 11.3

Repeat the multiplication of $P = Y \times X = 011001_2 \times 100111_2$ from Figure 11.71, applying Booth encoding to reduce the number of partial products.

SOLUTION: Figure 11.79 shows the multiplication. X is written vertically and the bits are used to select the four partial products. Each partial product is shifted two columns left of the previous one because it has four times the weight. The upper bits are sign-extended with 1s for negative partial products and 0s for positive partial products. The partial products are added to obtain the result.

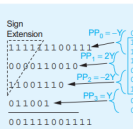


FIGURE 11.79 Booth-encoded exan

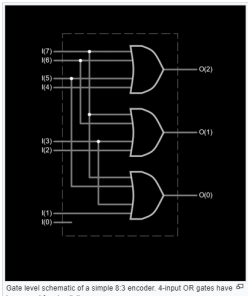
TABLE 4.4 Spreadsheet comparing decoder designs

Design	Stages N	G	P	D
NAND4-INV	2	2	5	29.8
NAND2-NOR2	2	20/9	4	30.1
INV-NAND4-INV	**3**	**2**	**6**	**22.1**
NAND4-INV-INV-INV	4	2	7	21.1
NAND2-NOR2-INV-INV	4	20/9	6	20.5
NAND2-INV-NAND2-INV	4	16/9	6	19.7
INV-NAND2-INV-NAND2-INV	5	16/9	7	20.4
NAND2-INV-NAND2-INV-INV-INV	6	16/9	8	21.6

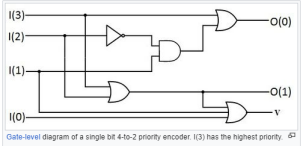
8-to-3 encoder [edit]

Truth Table

Input								Output		
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	O_2	O_1	O_0
0	0	0	0	0	0	0	0	x		
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1



Gate level schematic of a simple 8:3 encoder. 4-input OR gates have been used for simplicity.



Gate-level diagram of a single bit 4-to-2 priority encoder. I(3) has the highest priority.

A truth table of a single bit 4-to-2 priority encoder is shown, where the inputs are shown in decreasing order of priority left-to-right, and "x" indicates a don't care term - i.e. any input value there yields the same output since it is superseded by a higher-priority input. The (usually-included[4]) "V" output indicates if the input is valid.

4 to 2 Priority Encoder

Input				Output		
I_3	I_2	I_1	I_0	O_1	O_0	V
0	0	0	0	x		0
0	0	0	1	0	0	1
0	0	1	x	0	1	1
0	1	x		1	0	1
1	x			1	1	1