

EDA322: Digital Design

Re-exam August 2014

Date: Wednesday, August 28, 2014

Time: **14:00-18:00**

Examiner: Ioannis Sourdis

Department: Computer Science and Engineering

Inquiries: Ioannis Sourdis (extension 1744); will visit the room at **14:30** and at **16:30**

Results and grading review: See me in my office on **September 18th at 10:00-12:00**.

Exam Duration: 4 hours

Grading scale: 100 points in total

Chalmers:

0: 0%-49%, 3: 50%-64%, 4: 65%-84%, 5: 85%-100%

GU:

Fail (U): 0%-49%, Pass (G): 50%-79%, Pass with Distinction (VG): 80%-100%

Available references: Blank paper and a calculator are allowed. No textbooks or lecture notes, etc. allowed.

General: Submit your solutions, in English, on blank paper sheets. Write legibly; feel free to use figures to get your point across.

The order of answering the questions does not matter (start with the easiest ones).

Please start the solutions for each problem on a new sheet. Please number the sheets so that the solutions are in numerical order.

Note that it is possible to receive partial credit for an answer even if it is not 100% correct.

Your personal identity code is required on each submitted sheet!

Good luck!

Question 1: (10 points)

- a) Derive the canonical form of Sum of Products (SoP) for the following function $F = ab' + ab + a'bc'd$ (5 points)
- b) Then, derive the canonical form of the Product of Sums (PoS) for the function F. (3 points)
- c) What is a prime implicant and what is an essential prime implicant? (2 points)

Answer:

Can be solved in 3 different ways:

- 1. Making the truth table of the function and then finding the minterms (the sum of which is the SoP canonical form of F) and the maxterms (the product of which is the PoS of F). (similar to Lecture 2 slide 12)
- 2. Through Shannon Theorem (similar to Lecture 2 slide 16)
- 3. Expanding each product (similar to Lecture 2 slide 17)
- 4. After step 2. or 3. The PoS is derived (similar to Lecture 2 slide 12)

(Above mentioned lecture slides can be found at the end of this document)

a)

$$F = ab' + ab + a'bc'd = ab'c'd' + ab'c'd + ab'cd' + ab'cd + abc'd' + abc'd + abcd' + abcd + a'bc'd = \Sigma(5,8,9,10,11, 12,13,14,15)$$

b) $\Pi(0,1,2,3,4,6,7)$

c) Answer in Lecture 2, slide 30

(Above mentioned lecture slides can be found at the end of this document)

Question 2: (10 points)

Minimize the cost of function:

$$F(x_4, x_3, x_2, x_1, x_0) = \Sigma(9, 11, 13, 15, 23, 31) + D(8, 10, 12, 14, 19, 27)$$

using Quine-McCluskey. Measure the cost of the minimized function by counting the total number of 2-input gates of the circuit (e.g. $a*b + c*d$, has cost of 3).

Answer:

Truth table for the minterms
 $\Sigma(9, 11, 13, 15, 23, 31)$:

i	x4	x3	x2	x1	x0	y
0	0	0	0	0	0	0
1	0	0	0	0	1	0
2	0	0	0	1	0	0
3	0	0	0	1	1	0
4	0	0	1	0	0	0
5	0	0	1	0	1	0
6	0	0	1	1	0	0
7	0	0	1	1	1	0
8	0	1	0	0	0	0
9	0	1	0	0	1	1
10	0	1	0	1	0	0
11	0	1	0	1	1	1
12	0	1	1	0	0	0
13	0	1	1	0	1	1
14	0	1	1	1	0	0
15	0	1	1	1	1	1
16	1	0	0	0	0	0
17	1	0	0	0	1	0
18	1	0	0	1	0	0
19	1	0	0	1	1	0
20	1	0	1	0	0	0
21	1	0	1	0	1	0
22	1	0	1	1	0	0
23	1	0	1	1	1	1
24	1	1	0	0	0	0
25	1	1	0	0	1	0
26	1	1	0	1	0	0
27	1	1	0	1	1	0
28	1	1	1	0	0	0
29	1	1	1	0	1	0
30	1	1	1	1	0	0
31	1	1	1	1	1	1

Truth table for the don't care terms
 $D(8, 10, 12, 14, 19, 27)$:

i	x4	x3	x2	x1	x0	h*
0	0	0	0	0	0	0
1	0	0	0	0	1	0
2	0	0	0	1	0	0
3	0	0	0	1	1	0
4	0	0	1	0	0	0
5	0	0	1	0	1	0
6	0	0	1	1	0	0
7	0	0	1	1	1	0
8	0	1	0	0	0	1
9	0	1	0	0	1	0
10	0	1	0	1	0	1
11	0	1	0	1	1	0
12	0	1	1	0	0	1
13	0	1	1	0	1	0
14	0	1	1	1	0	1
15	0	1	1	1	1	0
16	1	0	0	0	0	0
17	1	0	0	0	1	0
18	1	0	0	1	0	0
19	1	0	0	1	1	1
20	1	0	1	0	0	0
21	1	0	1	0	1	0
22	1	0	1	1	0	0
23	1	0	1	1	1	0
24	1	1	0	0	0	0
25	1	1	0	0	1	0
26	1	1	0	1	0	0
27	1	1	0	1	1	1
28	1	1	1	0	0	0
29	1	1	1	0	1	0
30	1	1	1	1	0	0
31	1	1	1	1	1	0

The minterms and don't care terms are put together:

Step 1:

Indexgruppe	Index
0	
1	8 *
2	9 *
	10 *
	12 *
3	11 *
	13 *
	14 *
	19 *
4	15 *
	23 *
	27 *
5	31 *

step2:

Indexgruppe	Index
0/1	
1/2	9,8 (1) *
	10,8 (2) *
	12,8 (4) *
2/3	11,9 (2) *
	13,9 (4) *
	11,10 (1) *
	14,10 (4) *
	13,12 (1) *
	14,12 (2) *
3/4	15,11 (4) *
	27,11 (16) *
	15,13 (2) *
	15,14 (1) *
	23,19 (4) *
	27,19 (8) *
4/5	31,15 (16) *
	31,23 (8) *
	31,27 (4) *

step3:

Indexgruppe	Index
0/1/2	
1/2/3	11,10,9,8 (1,2) *
	13,12,9,8 (1,4) *
	11,9,10,8 (2,1) *
	14,12,10,8 (2,4) *
	13,9,12,8 (4,1) *
	14,10,12,8 (4,2) *
2/3/4	15,13,11,9 (2,4) *
	15,11,13,9 (4,2) *
	15,14,11,10 (1,4) *
	15,11,14,10 (4,1) *
	15,14,13,12 (1,2) *
	15,13,14,12 (2,1) *
3/4/5	31,27,15,11 (4,16) P1
	31,15,27,11 (16,4)
	31,27,23,19 (4,8) P2
	31,23,27,19 (8,4)

step 4:

Indexgruppe	Index
0/1/2/3	
1/2/3/4	15,14,13,12,11,10,9,8 (1,2,4) P3
	15,14,11,10,13,12,9,8 (1,4,2)
	15,13,14,12,11,9,10,8 (2,1,4)
	15,13,11,9,14,12,10,8 (2,4,1)
	15,11,14,10,13,9,12,8 (4,1,2)
	15,11,13,9,14,10,12,8 (4,2,1)
2/3/4/5	

$$P1 = x_3 x_1 x_0$$

$$P2 = x_4 x_1 x_0$$

$$P3 = x_4' x_3$$

P/M1	9	11	13	15	23	31
P1		*		*		*
P2					*	*
P3	*	*	*	*		

$$F = P2 + P3 = x_4 x_1 x_0 + x_4' x_3$$

Cost is 4.

Question 3: (10 points)

Draw the block diagram of a 4-bit unsigned binary (Sequential) Divider, show its 3 registers (\$M, \$R, \$Q) and explain their use. Explain step-by step the division of 6/4 (6=Dividend, 4=Divisor) on the above divider.

Answer:

Lecture 11, slide 24 (for 4 bits instead of 32-bits), slide 22, and example of slide 23.

(Above mentioned lecture slides can be found at the end of this document)

Question 4: (10 points)

Use partitioning to minimize the number of states in the following Finite State Machine (FSM).

Next State / Output

Present State	Input= 0	Input=1
A	C/1	B/0
B	C/1	E/0
C	B/1	E/0
D	D/0	B/1
E	E/0	A/1

- a) *Minimize the number of state using partitioning. (8 points)*
- b) *Create the state table of the minimized version of the FSM. (2point)*

Answer:

a)

$$\begin{array}{l} P_1=(A,B,C) (D,E) \\ X=0 \ C \ C \ B \ D \ E \} \\ X=1 \ B \ E \ E \ B \ A \} \end{array}$$

$$\begin{array}{l} P_2=(A) (B,C) (D,E) \\ X=0 \ \ C \ B \ D \ E \} \\ X=1 \ \ E \ E \ B \ A \} \end{array}$$

$$\begin{array}{l} P_3=(A) (B,C) (D) (E) \\ X=0 \ \ \ C \ B \\ X=1 \ \ \ E \ E \end{array}$$

$$\begin{array}{l} P_4=(A) (B, \cancel{C}) (D) \\ (E) = P_3 \end{array}$$

b)

	0	1
A	B/1	B/0
B	B/1	E/0
D	D/0	B/1
E	E/0	A/1

Question 5: (10 points)

- a) Draw a typical FPGA Logic Cell with 4-bits input and 1-bit output, which is optionally registered.
- b) What are the values of the configuration bits when configured to implement:
 - i. An AND gate with registered output;
 - ii. An OR gate with non-registered output.

Answer:

- a) Lecture 12, slide 18
(Above mentioned lecture slides can be found at the end of this document)
- b)
 - i. all LUT entries 0 except the one at location "1111", output mux selecting the flip-flop output
 - ii. all LUT entries 1 except the one at location "0000", output mux selecting the LUT output

Question 6: (10 points)

Draw the block diagram of a Memory block with 4 rows (memory entries) of 3-bits each. Explain how reading and writing to a memory-entry is performed.

Note: the block diagram has to be at gate-level, except flip-flops, which can be drawn as boxes without showing their internal gates.

Answer:

Lecture 11, Slides 34, 35

(Above mentioned lecture slides can be found at the end of this document)

Question 7: (10 points)

Draw an SR-latch, describe its State Transition Diagram and its Timing diagram.

Answer:

Lecture 4 slides 8, 9, 10

(Above mentioned lecture slides can be found at the end of this document)

Question 8: (10 points)

- a) Draw the block diagram of a 4-bit ripple-carry adder (showing each gate separately even inside a full adder). (2 points)
- b) Show the critical path of the ripple carry adder and calculate its delay considering that a XOR gate has a delay of 2 nsec, an AND gate has a delay of 1 nsec and an OR gate a delay of 1 nsec. (5 points)
- c) What would be the critical path if the inputs and outputs of the adder are registered with flip-flops that have a setup time of 0.5 nsec and a propagation time 0.5 nsec? (3 points)

Answer:

a) Lecture 2, slide 73.

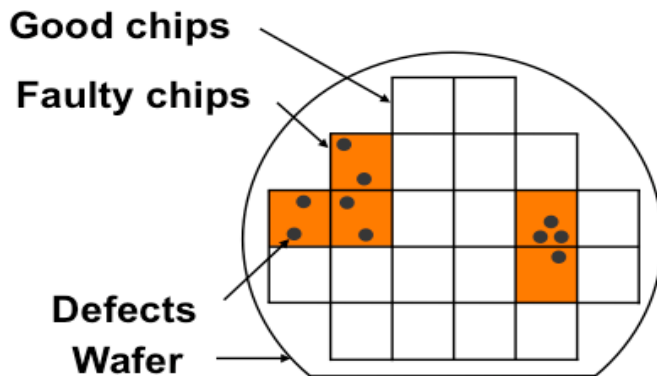
(Above mentioned lecture slides can be found at the end of this document)

b) Delay = $(2+1+1) + (1+1) + (1+1) + (2) = 10$ nsec

c) Delay = $10 + 0.5 + 0.5 = 11$ nsec

Question 9: (10 points)

What is the yield for the following wafer:



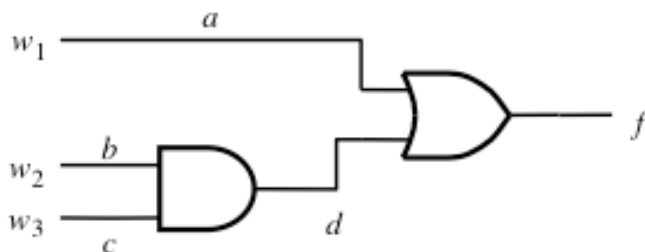
Answer:

Lecture 13, slide 10

(Above mentioned lecture slides can be found at the end of this document)

Question 10: (10 points)

Find the test-set of inputs that detect all the single stuck-at faults for the following circuit:



Answer:

Lecture 13, slide 24.

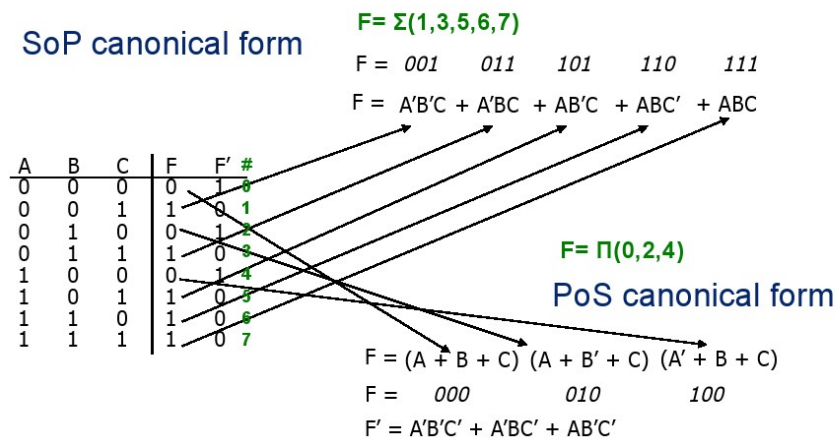
(Above mentioned lecture slides can be found at the end of this document)

END of the EXAM

Lecture Slides

Question 1:

Canonical Forms



canonical form \neq minimal form

EDA322 Digital Design,
2015-2016, Lecture 2

I. Sourdis, CSE, Chalmers

12

Conversion between canonical forms

Relation between maxterms and minterms of a function

$$F(x,y,z) = \Sigma(1,3,5,6,7) = \Pi(0,2,4)$$

... and for the inverted function:

$$F'(x,y,z) = \Pi(1,3,5,6,7) = \Sigma(0,2,4)$$

EDA322 Digital Design,
2015-2016, Lecture 2

I. Sourdis, CSE, Chalmers

13

Shannon's expansion theorem

- **Shannon's expansion theorem**

(a). $f(x_1, x_2, \dots, x_n) = x_1 * f(1, x_2, \dots, x_n) + x_1' * f(0, x_2, \dots, x_n)$

(b). $f(x_1, x_2, \dots, x_n) = [x_1 + f(0, x_2, \dots, x_n)] [x_1' + f(1, x_2, \dots, x_n)]$

Example:

$$\begin{aligned} F(x_1, x_2, x_3) &= x_1x_2 + x_1x_3 + x_2x_3 \\ &= x_1'F(0, x_2, x_3) + x_1F(1, x_2, x_3) \\ &= x_1' (0x_2 + 0x_3 + x_2x_3) + x_1(1x_2 + 1x_3 + x_2x_3) \\ &= x_1' (x_2x_3) + x_1(x_2 + x_3 + x_2x_3) \\ &= x_1' (x_2x_3) + x_1(x_2 + x_3(1 + x_2)) \\ &= x_1' (x_2x_3) + x_1(x_2 + x_3) \end{aligned}$$

Derivation of Canonical Forms

- Derive canonical PoS or SoP using Boolean algebra.

- **Shannon's expansion theorem**

(a). $f(x_1, x_2, \dots, x_n) = x_1 * f(1, x_2, \dots, x_n) + x_1' * f(0, x_2, \dots, x_n)$

(b). $f(x_1, x_2, \dots, x_n) = [x_1 + f(0, x_2, \dots, x_n)] [x_1' + f(1, x_2, \dots, x_n)]$

Convert a function to SoP

1. Apply Shannon Theorem until you get sum of minterms
2. Get the function to a SoP form, as follows:
 - if a product of literals is a minterm, keep it
 - for every variable x_i that doesn't exist in a product we add $(x_i + x_i')$,
e.g.:
 - $x_1 * x_2 = x_1 * x_2 * (x_3 + x_3') = x_1 * x_2 * x_3 + x_1 * x_2 * x_3'$
 - Continue and delete redundant products

Derivation of Canonical Forms

- Derive canonical PoS or SoP using Boolean algebra.
- **Shannon's expansion theorem**
 - (a). $f(x_1, x_2, \dots, x_n) = x_1 * f(1, x_2, \dots, x_n) + x_1' * f(0, x_2, \dots, x_n)$
 - (b). $f(x_1, x_2, \dots, x_n) = [x_1 + f(0, x_2, \dots, x_n)] [x_1' + f(1, x_2, \dots, x_n)]$
- **Example:** $f(A,B,C) = AB + AC' + A'C$
 - $f(A,B,C) = AB + AC' + A'C = A f(1,B,C) + A' f(0,B,C)$
 $= A(1 \cdot B + 1 \cdot C' + 1 \cdot C) + A'(0 \cdot B + 0 \cdot C' + 0 \cdot C) = A(B + C') + A'C$
 - $f(A,B,C) = A(B + C') + A'C = B[A(1+C')] + A'C + B'[A(0 + C') + A'C]$
 $= B[A + A'C] + B'[AC' + A'C] = AB + A'BC + AB'C' + A'B'C$
 - $f(A,B,C) = AB + A'BC + AB'C' + A'B'C$
 $= C[AB + A'B \cdot 1 + AB' \cdot 1' + A'B' \cdot 1] + C'[AB + A'B \cdot 0 + AB' \cdot 0' + A'B' \cdot 0]$
 $= ABC + A'BC + AB'C' + ABC' + AB'B'C$

Derivation of Canonical Forms

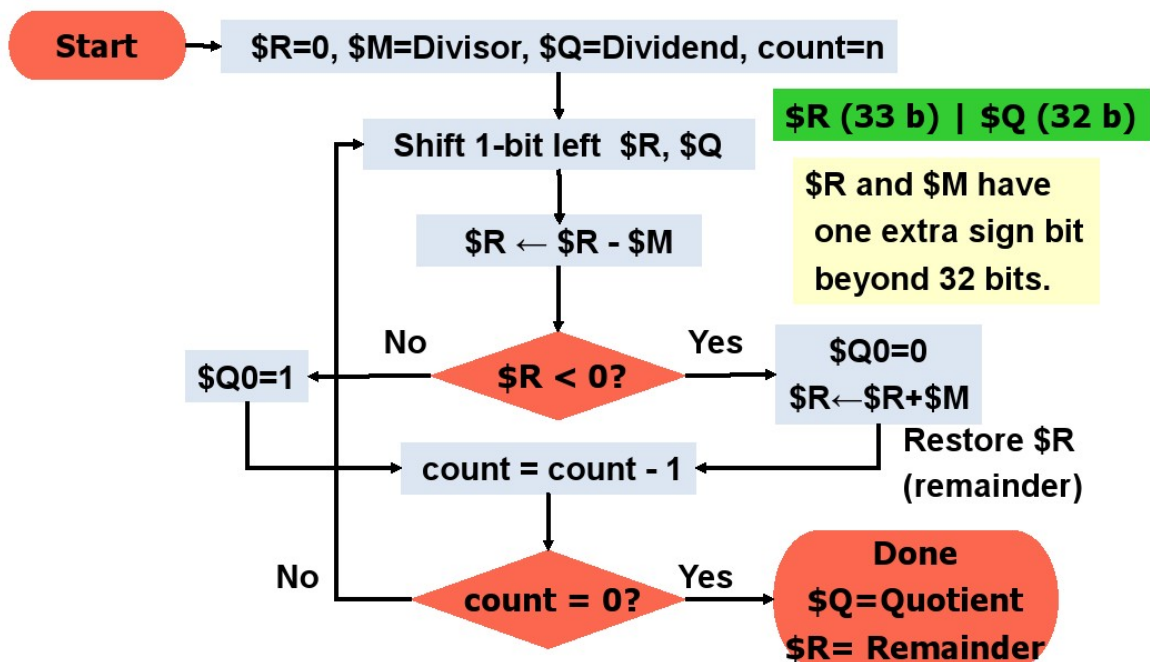
- Derive canonical PoS or SoP using Boolean algebra.
- **Shannon's expansion theorem**
 - (a). $f(x_1, x_2, \dots, x_n) = x_1 * f(1, x_2, \dots, x_n) + x_1' * f(0, x_2, \dots, x_n)$
 - (b). $f(x_1, x_2, \dots, x_n) = [x_1 + f(0, x_2, \dots, x_n)] [x_1' + f(1, x_2, \dots, x_n)]$
- **Example:** $f(A,B,C) = AB + AC' + A'C$
 - $f(A,B,C) = AB + AC' + A'C =$
 $= AB(C+C') + AC'(B+B') + A'C(B+B')$
 $= ABC + ABC' + ABC' + AB'C' + A'BC + A'B'C$
 $= ABC + ABC' + AB'C' + A'BC + A'B'C$

Definition of terms

- A function F **covers** a function g , if it takes the value '1' when function g does.
- **Implicant**: a product of literals of a function f for which f gets the value '1'.
- **Prime implicant**: an implicant that cannot be covered by a more general (more reduced - meaning with fewer literals) implicant.
- **Essential prime implicant**: a prime implicant of function f that includes a minterm, not included by any other prime implicant of the function.

Question 3:

32-bit Binary Division Flowchart



count

4-bit Example: $6/4 = 1$, Remainder 2

	Actions	n	\$R, \$Q	\$M = Divisor
4	Initialize	4	00000 0110	00100
	Shift left \$R, \$Q	4	00000 1100	00100
	Add - \$M (11100) to \$R	4	11100 1100	00100
3	Restore, add \$M (00100) to \$R	3	00000 1100	00100
	Shift left \$R, \$Q	3	00001 1000	00100
	Add - \$M (11100) to \$R	3	11101 1000	00100
2	Restore, add \$M (00100) to \$R	2	00001 1000	00100
	Shift left \$R, \$Q	2	00011 0000	00100
	Add - \$M (11100) to \$R	2	11111 0000	00100
1	Restore, add \$M (00100) to \$R	1	00011 0000	00100
	Shift left \$R, \$Q	1	00110 0000	00100
	Add - \$M (11100) to \$R	1	00010 0000	00100
0	Set LSB of \$Q = 1	0	00010 0001	00100

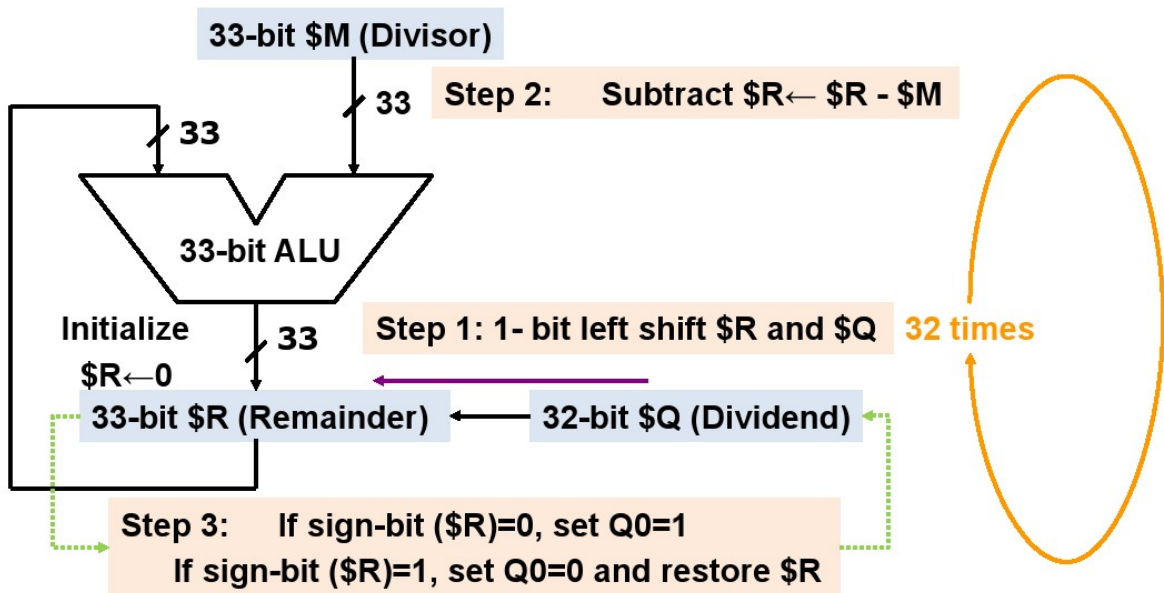
Remainder | Quotient

EDA322 Digital Design,
2015-2016, Lecture 11

I. Sourdis, CSE, Chalmers

23

Division



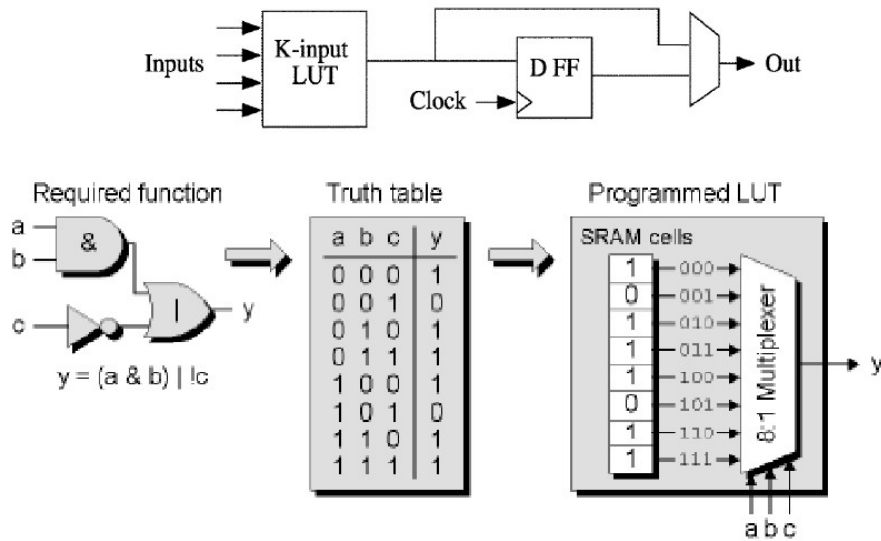
EDA322 Digital Design,
2015-2016, Lecture 11

I. Sourdis, CSE, Chalmers

24

Question 5:

Mapping logic to a Logic Cell

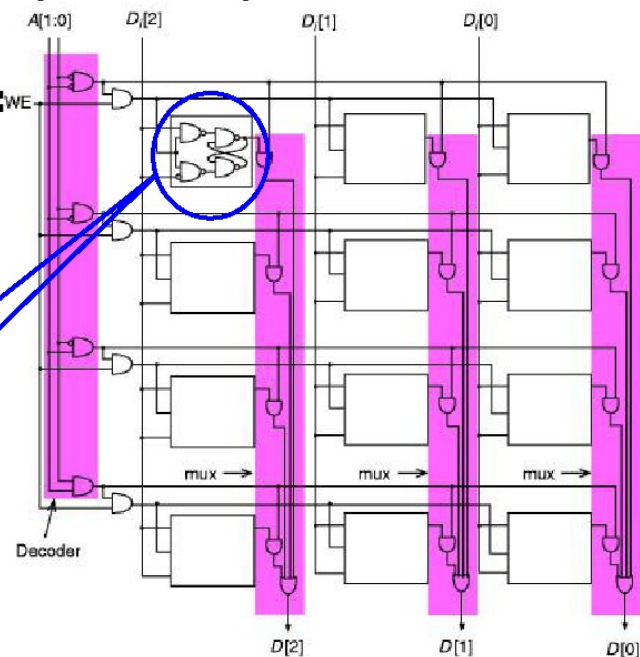


Question 6:

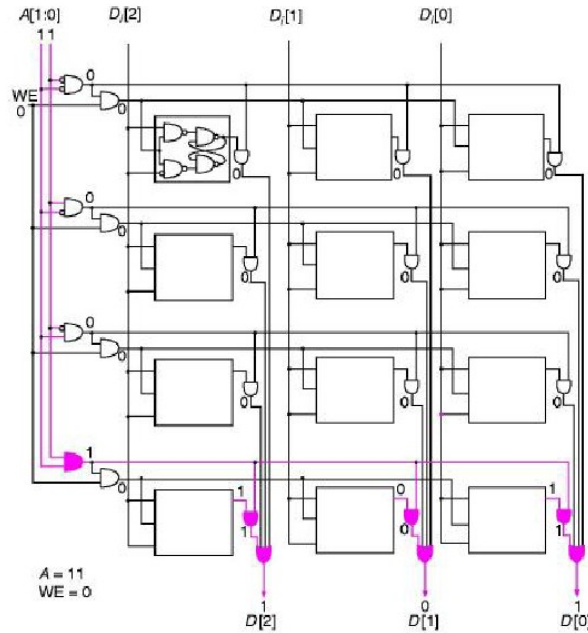
Memory Example

A 2² by 3 bits memory:
 two address lines: A[1:0]
 three data lines: D[2:0]
 one control line: WE

One gated D-latch

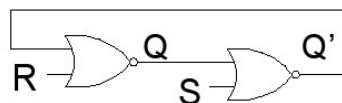


Reading a location in memory

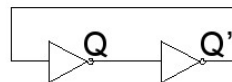


Question 7:

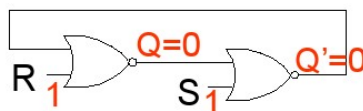
Cross-coupled NOR gates



- If both $R=0$ & $S=0$, then cross-coupled NORs equivalent to a stable latch:

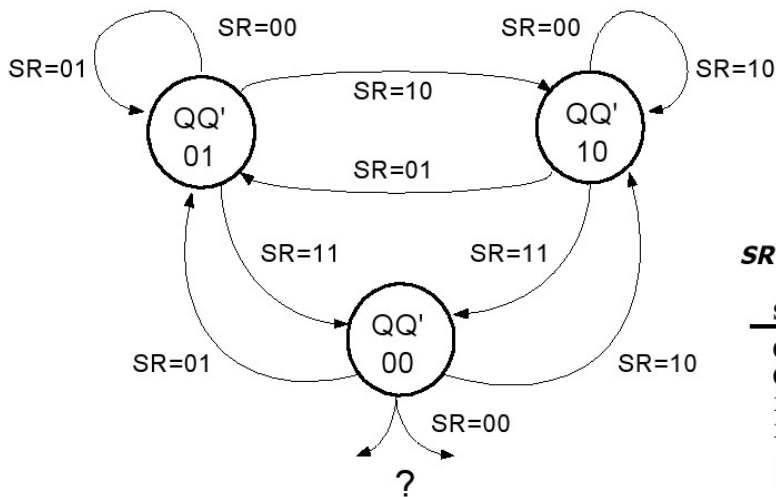


- What happens if R or S or both become = 1?



NOR	
00	1
01	0
10	0
11	0

Asynchronous State Transition Diagram

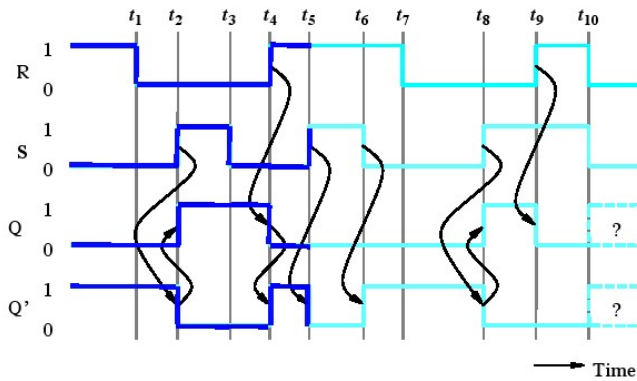


SR Latch:

S	R	Q	Q'	
0	0	0/1	1/0	(no change)
0	1	0	1	
1	0	1	0	
1	1	0	0	

(b) Truth table

SR-latch timing diagram

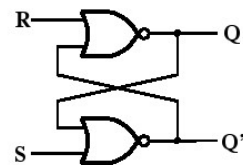


(c) Timing diagram

SR Latch:

S	R	Q	Q'	
0	0	0/1	1/0	(no change)
0	1	0	1	
1	0	1	0	
1	1	0	0	

(b) Truth table



(a) Circuit

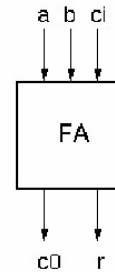
Question 8:

Ripple-Carry Adder

- Each cell:

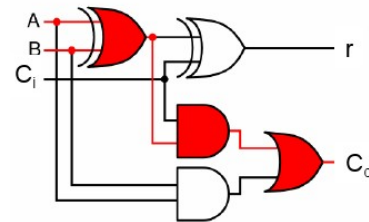
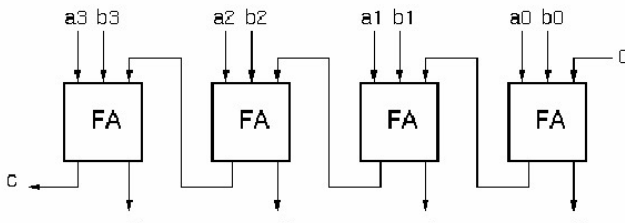
$$r_i = a_i \text{ XOR } b_i \text{ XOR } c_{in}$$

$$c_{out} = c_{in}(a_i \text{ XOR } b_i) + a_i b_i$$



“Full adder cell”

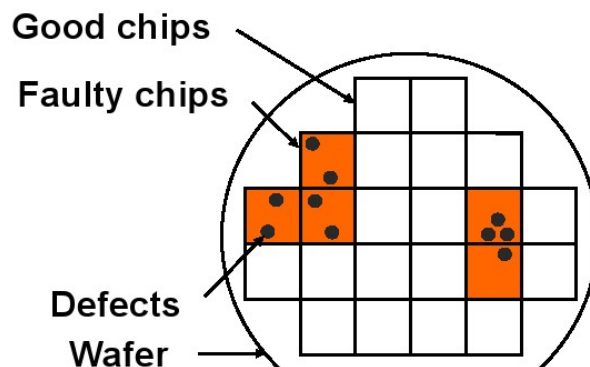
- 4-bit adder:



- What about subtraction?

Question 9:

Circuit Fabrication and Defects



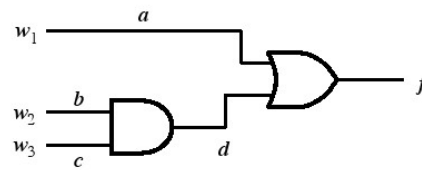
Defects are permanent faults caused during the Manufacturing process

$$\text{yield} = \frac{\text{number of working chips produced}}{\text{Total number of chips produced}}$$

$$\text{Wafer yield} = 17/22 = 0.77$$

Question 10:

Testset



One possible approach but unrealistic for big circuits!

Test $w_1 w_2 w_3$	Fault detected									
	a/0	a/1	b/0	b/1	c/0	c/1	d/0	d/1	f/0	f/1
000		✓						✓		
001		✓		✓				✓		✓
010		✓				✓		✓		✓
011			✓		✓		✓		✓	
100	✓								✓	
101	✓								✓	
110	✓								✓	
111									✓	