E4.20
AC3
ISE4.19

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2006

MSc and EEE/ISE PART IV: MEng and ACGI

CORRECTED COPY

INTRODUCTION TO DIGITAL INTEGRATED CIRCUIT DESIGN

Tuesday, 2 May 10:00 am

Time allowed: 3:00 hours

There are SIX questions on this paper.

Answer FOUR questions.

All questions carry equal marks

Any special instructions for invigilators and information for candidates are on page 1.

Examiners responsible First Marker(s) : K. Masselos
Second Marker(s) : P.Y.K. Cheung

Special instructions for invigilators

Students may bring any written or printed aids into the examination.

Special instructions for students

Students may need red, green, blue, yellow and black coloured pens.

The Questions

1. a) State five different implementation styles for digital integrated circuits and their fabrication processes.

[3]

b) Discuss the implementation options above in terms of implementation efficiency (transistor density per unit area, performance) and development time (including both design time and fabrication time). Give their relative position in the 2-D space (development time, implementation efficiency) in Figure 1.1.

[10]

c) Discuss the following issues related to verification:

i) With the help of an example, explain the main trade-off related to simulation.

[3]

ii) What is a false timing path?

[2]

d) Why is hardware/software co-design part of a modern integrated circuit design flow? What is the basic advantage that high level synthesis could bring compared to conventional design approaches?

[2]

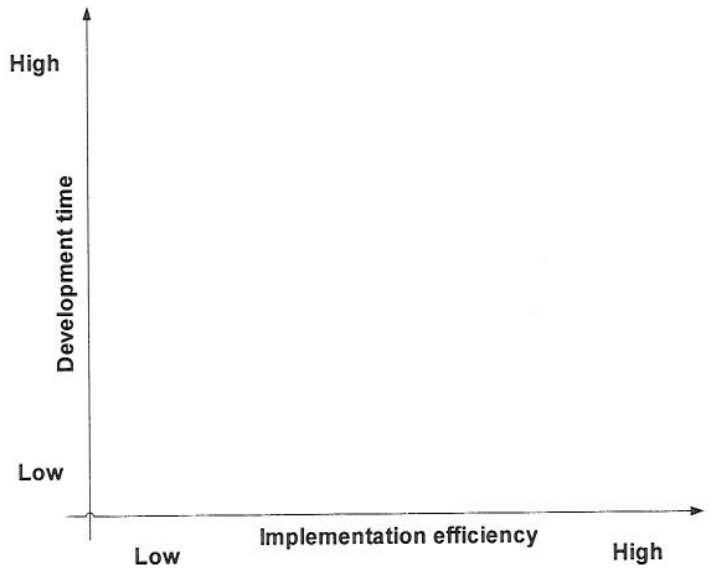


Figure 1.1 Development time – implementation efficiency space

2. a) Implement the following Boolean function as a single compound/complex CMOS static gate.

$$\overline{A \bullet (B + C) + D \bullet E}$$

[5]

b) Use any layout representation (e.g. stick diagram) to design the layout of the circuit.

[5]

c) Implement the following Boolean function using pseudo-NMOS and discuss the advantages and disadvantages of pseudo-NMOS compared to static CMOS.

$$A = (B \bullet C) + (D \bullet E)$$

[5]

d) Explain analytically why nMOS and pMOS devices cannot pass efficiently logic 1 and logic 0 respectively.

[5]

3. a) Figure 3.1 (see the colour supplementary sheet) shows the layout of a CMOS cell with four inputs and four outputs. Extract and draw the transistor level schematic diagram.

[12]

b) What function does the circuit perform?

[5]

c) State two major characteristics of the circuit topology.

[3]

4. a) Explain the main difference between SRAM and DRAM cells and the relevant trade-off.

[4]

b) Describe one basic reason for the lower performance of DRAMs compared to SRAMs. What is the major problem for a DRAM from a system point of view ?

[4]

c) Using simple assumptions for the inputs, derive analytically the energy consuming transition probability for a three input NOR gate in static and dynamic logic. Comment on the results.

[6]

d) Derive equations for the energy consuming transition probability of 3 input NAND and NOR static and dynamic CMOS gates as a function of input transition probabilities.

[6]

5. a) Design a dynamic circuit to implement the following Boolean function:

$$Z = \overline{A \bullet (B + C) + (D \bullet E)}$$

[5]

b) Discuss the principle of operation of dynamic logic and its basic advantages over static CMOS and ratioed logic.

[5]

c) Using the circuit in Figure 5.1, explain the main problem of dynamic logic.

[5]

d) Describe how the domino logic circuit in Figure 5.2 overcomes this problem.

[5]

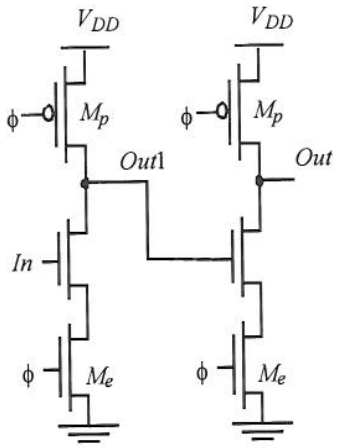


Figure 5.1 Dynamic logic circuit

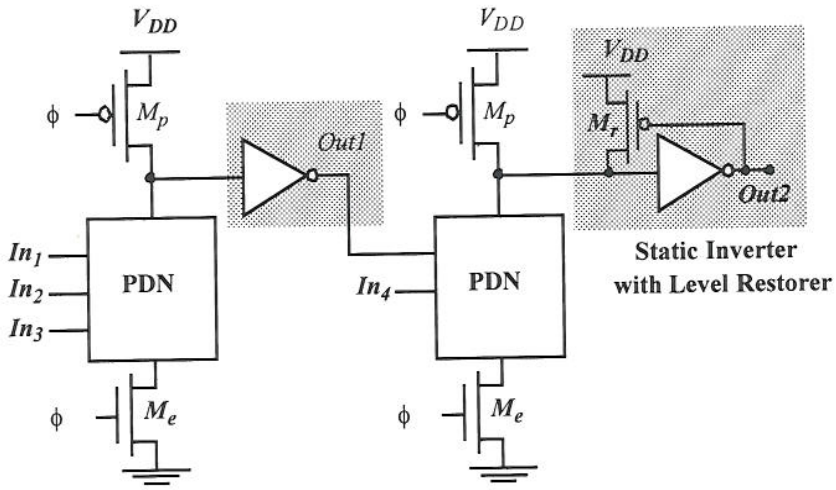


Figure 5.2 Domino logic circuit

6. a) Discuss the principles of the stuck-at 0/1 fault model. What type of faults does it explicitly cover and which not? What do we gain and what do we lose by not including all types of faults in the model? Why is this model the most popular?

[4]

b) Derive the minimum set of test vectors for detecting stuck-at faults in the inputs of the circuit shown in Figure 6.1.

[10]

c) Generate test vectors for detecting the stuck-at zero and stuck-at one faults in nodes E and F respectively of the circuit shown in Figure 6.2.

[6]

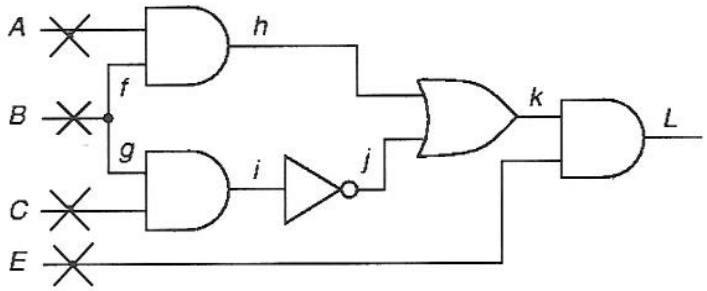


Figure 6.1

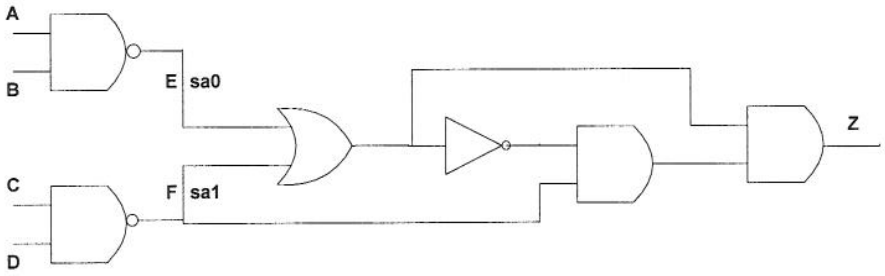


Figure 6.2

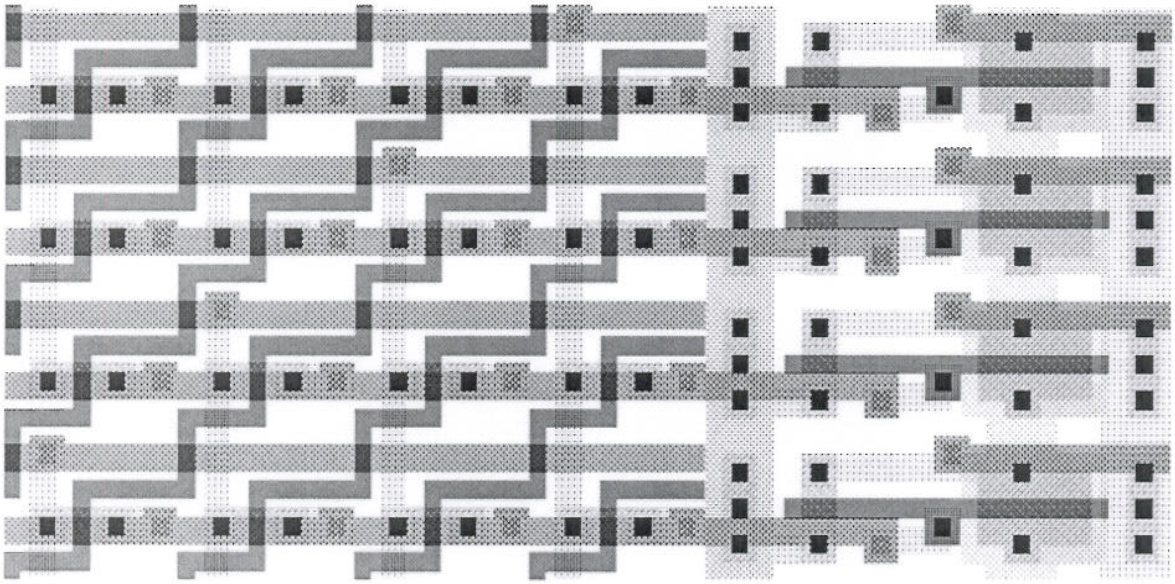


Figure 3.1 Layout of a full custom cell for Question 3

Master - 20. 4. 06

Introduction to Digital IC design 05/06

E4.20/ACJ/Jc 4.19

The Solutions

1. a) **Full custom design** – requires full run through the fabrication process

Semi custom/standard cells design – requires full run through the fabrication process

Macro cell based design – requires full run through the fabrication process

Pre-diffused gate array based design – transistors are predefined on the wafer, customization is achieved by adding the desired interconnections with only a few metallization steps

Pre-wired arrays - No dedicated manufacturing stages altogether, pre-processed die that can be programmed on the field (outside the semiconductor foundry)

Bookwork [3]

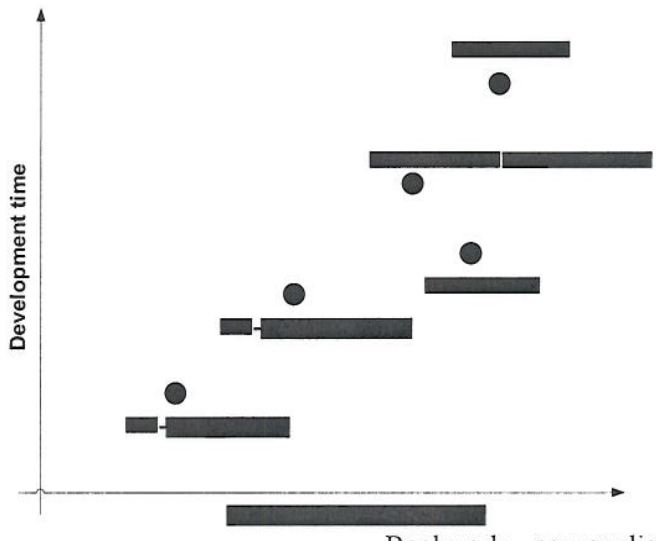
b) **Full custom design**: Highest implementation efficiency and development time

Semi custom design: Decreased implementation efficiency and development time (design time) compared to full custom design due to the use of pre-designed library cells

Macro cell based design: For functions of specific nature macro cells offer better implementation efficiency and shorter development time (design time) than standard cells (assuming that the macrocells are designed only once and then re-used)

Pre-diffused gate array based design: Lower implementation efficiency and shorter development time than semi custom/standard cells

Pre-wired arrays: Lower implementation efficiency and shorter development time than pre-diffused gate array based design



Bookwork – new application of theory [10]

c)

i) Accuracy versus simulation speed. Circuit level simulation (e.g. SPICE) versus switch level simulation (e.g. IRSIM) that uses a simplified transistor model.

Bookwork [2.5]

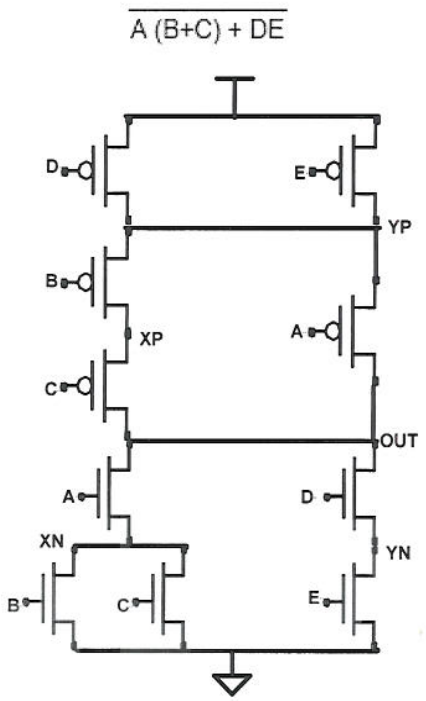
ii) A critical path that can never be exercised during normal circuit operation

Bookwork [2.5]

d) Because modern integrated circuits (SoCs) include CPUs alongside custom logic. High level synthesis could potentially increase IC design productivity by raising the level of abstraction.

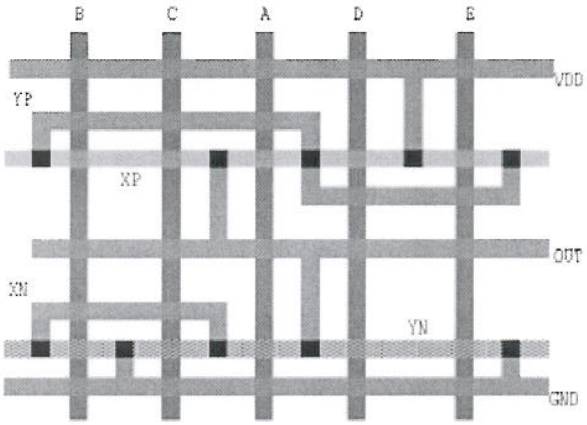
Bookwork [2]

2. a)



New application of theory [5]

b)

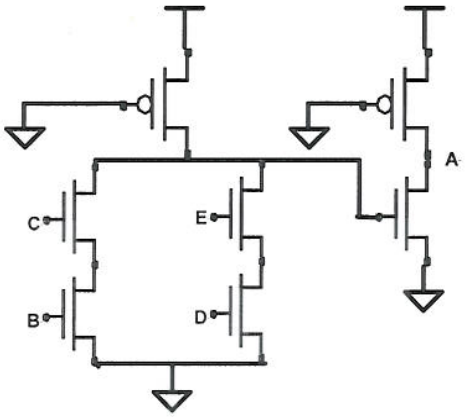


New application of theory [5]

c) Advantage: smaller number of transistors (N+1 instead of 2N)

Disadvantages: static power dissipation, slower rise time

A= (B·C) + (D·E)



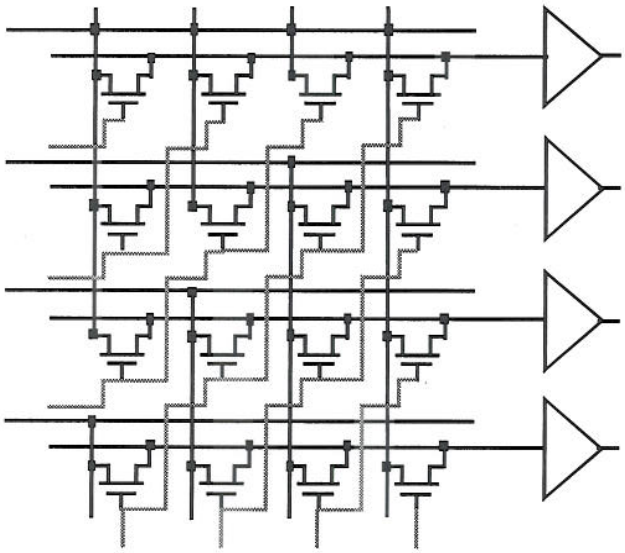
New application of theory – bookwork [5]

d) nMOS: Assume nMOS transistor with the gate and drain tied to Vdd and Vs originally 0. $V_{gs} > V_{tn}$ so the transistor is ON and current flows. If the voltage on the source rises to $V_s = V_{dd} - V_{tn}$, V_{gs} falls to V_{tn} and the transistor cuts itself OFF. Therefore, nMOS transistors attempting to pass a '1' never pull the source above $V_{dd} - V_{tn}$.

pMOS: Assume pMOS transistor with the gate and drain tied to Gnd. If the pMOS source drops below the absolute value of V_{tp}, the transistor cuts OFF. Thus, pMOS transistors only pull down to within a threshold above Gnd.

Bookwork [5]

3. a)



New application of theory [12]

b) 4 inputs Barrel shifter with 4 stages

Bookwork [5]

c) Area is dominated by wiring

A control wire is needed for every shift bit

Bookwork [3]

4. a) The main difference between SRAM and DRAM cells is the way they preserve the charge stored in the cell. SRAM cells use extra pull-up transistors to produce a logic "1" while DRAM cells store charge in capacitances and use the refresh mechanism to compensate for leakage. As a result SRAM cells are larger and faster than DRAM cells.

Bookwork [4]

b) DRAMs use multiplexed addressing (several cycles to input the address) and complex interface signalling (with large number of signals) while SRAMs are self timed (timing signals are generated internally based on the address) and require one cycle to receive their address. The problem with DRAMs is the speed gap between them and the CPUs.

Bookwork [4]

c) Assumption for both cases: P(A=1)=P(B=1)=P(C=1)=1/2

Static case – energy consuming output transition: 0 to 1.

P(Out=1)=P(A=0)P(B=0)P(C=0)=1/8

P(Out=0)=1-P(Out=1)=7/8

Probability of energy consuming output transition=P(Out=0)P(Out=1)=7/64

Dynamic case – energy consuming output transition: Assuming precharge phase first (Out=1) energy is consumed whet Out is evaluated to 0

P(Out=0)=7/8 (for all input combinations with at least one input equal to 1).

Switching activity probability of static is lower than dynamic

New computed example [6]

d) Assuming that P_A, P_B, P_C are the probabilities that A, B and C are 1 then:

Static 0 to 1 transition:

NAND: $P_A P_B P_C (1- P_A P_B P_C)$

NOR: $(1- P_A) (1-P_B) (1-P_C) (1-(1- P_A) (1-P_B) (1-P_C))$

Dynamic from precharge to 1 to 0:

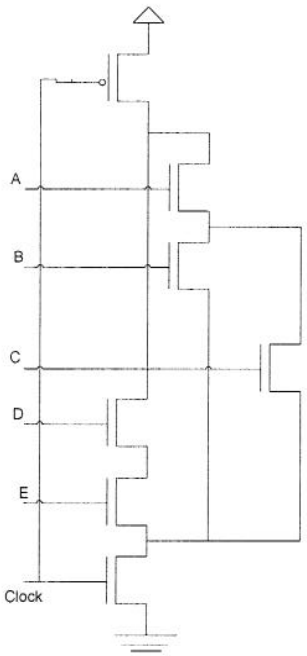
NAND: $P_A P_B P_C$

NOR: $1-(1- P_A) (1-P_B) (1-P_C)$

New theory [6]

5. a)

$$\mathbf{Z = \overline{A \cdot (B+C) + (D \cdot E)}}$$



New application of theory [5]

b) The clock signal is used to divide the gate operation into two halves. In the first half, the output node is pre-charged to a high or low logic state. In the second half of a clock cycle, the circuit evaluates the correct output state.

Smaller number of transistors than static CMOS – no static power dissipation as ratioed logic

Bookwork [5]

c) Problem with cascading dynamic circuits: Inputs can only be changed when Ø (clock) is low and must be stable when Ø (clock) is high. When Ø is low, both P1 and P2 are precharged to a high voltage. However when Ø is high, delay through on the output P1 may erroneously discharge P2.

Bookwork [5]

d) The extra inverter ensures that the output is low during precharge, and prevents the next stage from evaluating, until the current stage has finished evaluation. This ensures that each stage (at the output of the inverter) will make at most a single transition from 0 to 1.

Bookwork [5]

6. a) Considers only the short circuits to GND and Vdd – stuck at 0 (sa0), stuck at 1 (sa1). It does not cover stuck-at-open and stuck-at-short faults. If these are also introduced the test pattern generation process becomes complicated. Moreover a large number of these faults are covered by the sao-sa1 model. So it can be said that it works reasonably well. The three sub-questions of Q1 a) cover the principles of the stuck-at 0/1 model. The faults covered by the stuck-at 0/1 model are short circuits to GND and VDD. Faults not covered include stuck-at-open and

stuck-at-short. In second sub-question the test pattern generation procedure becomes less complicated at the expense of smaller fault coverage. In third sub-question stuck-at 0/1 model is popular because stuck-at 0/1 faults in many cases can be used to model other types of faults thus offering good coverage at reasonable pattern generation complexity.

Bookwork [4]

b) Derive the minimum set of test vectors for detecting stuck-at faults in the inputs of the circuit shown in Figure 6.1.

Input	sa0	sa1
A	1111 (0 instead of 1)	0111 (1 instead of 0)
B	0111 (1 instead of 0)	0011 (0 instead of 1)
C	0111 (1 instead of 0)	0101 (0 instead of 1)
E	1111 (0 instead of 1) many other options but we select this to end up with minimum set of vectors)	1110 (1 instead of 0)

New computed example [10]

c) Generate test vectors for detecting the stuck-at zero and stuck-at one faults in nodes E and F respectively of the circuit shown in Figure 6.2.

There are no vectors to detect the two faults. This is because of the reconvergent fan out which keeps output to 0 for whatever combination of the inputs.

New computed example [6]