# Work for Digital Electronics Supervision II

The questions below are partitioned into two sets: the *core* questions and the *extension* questions. The core questions are based on material that the course should have covered by the end of lecture 6, material with which you should be familiar. Nevertheless, they will likely take a bit of thought. This supervision features the greatest number of core questions: start them early!

The extension questions are based on the same material, but will take longer, and require a lot more thought. Attempting them is optional. The more of the extension questions that you try, the better you will become at Digital Electronics and the easier you will find the work in future weeks. However, do not sacrifice your progress in other subjects or your health for the sake of completing them.

Please attempt all nine core items and as many of the extension questions as you can, and hand in your work to me by 1800 on Thursday, 22nd October. Either typeset your answers, working and any additional thoughts in TEX or LATEX and email a PDF to `mti20@cam.ac.uk`, or deposit your handwritten work in my Churchill pigeonhole.

## Core questions

1. Examples sheet questions 8, 13 [AM], 14 [AM], and 16.

   For 13, draw the circuit diagram of the system described before attempting the question. In addition to binary and denary, also write out the output sequence in octal and hexadecimal. Do persevere; the result is satisfying.

   In 14, do ensure that your transition logic is minimal.

   For 16, also choose encodings for each state, write them on the state diagram, and explain the reasons behind your choice of encodings.

2. A number of Boolean functions conform to the map shown below:

   | B \ A | 0 | 1 | X | 0 |
   |---|---|---|---|---|
   | | X | X | 1 | 1 |

   C

   Exhibit maps and minimum sum-of-product expressions for the following particular functions:

   (a) $f_1$, being a function of two variables

   (b) $f_2$, having four minterms

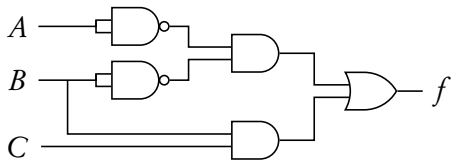
   (c) $f_3$, having two prime implicants

   (d) $f_4$, having a prime implicant that cannot appear in the minimum sum-of-products representation.

3. Design a circuit that implements an eight-bit binary multiplier. If you wish, you may abstract to large functional blocks such as "1-bit left shift" and "$n$-bit adder", instead of giving the low-level circuit details in full.

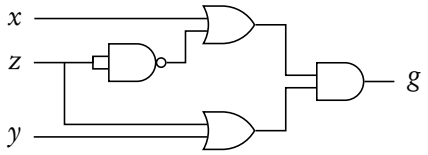
4. For each of the following circuits, write down the propagation delay and contamination delay, and draw a timing diagram showing the output of each gate when the inputs change as indicated. Each gate has a propagation delay of $\tau$. If a static hazard exists, identify its type (static 1 or static 0), and propose a solution. Draw your amended circuit, that does not suffer from the glitch.

(i)




$A$ is a constant 0, $C$ is a constant 1, and $B$ switches from 1 to 0.

(ii)




$x$ and $y$ are constant 0s. $z$ is initially 0, switches to 1 for a time $t \gg \tau$, and then returns to 0.

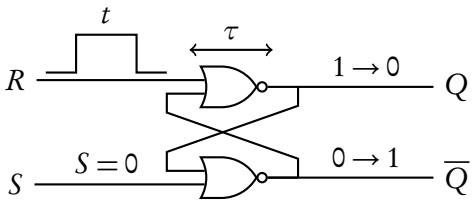
What name is commonly given to the function $g$?

You may find it helpful to complete the timing diagrams on 5mm squared paper.

5. (a) What is the difference between a flip flop and a latch? How might a flip flop be constructed from a latch? Give an example (or two) of the use of each, including timing diagrams to show how and when the signals change.

   (b) How might one use an SR latch to debounce a SPDT switch? Draw timing diagrams to illustrate your answer.

   (c) Suppose a NOR gate has a propagation delay equal to $\tau$. Further suppose that a particular SR latch is in state $Q = 1$, and a clean pulse of length $t$ appears on the $R$ input in order to reset the latch to state $Q = 0$. What is the minimum possible $t$, expressed in terms of $\tau$, in order for the intended state change to be effected? Support your answer with timing diagrams.



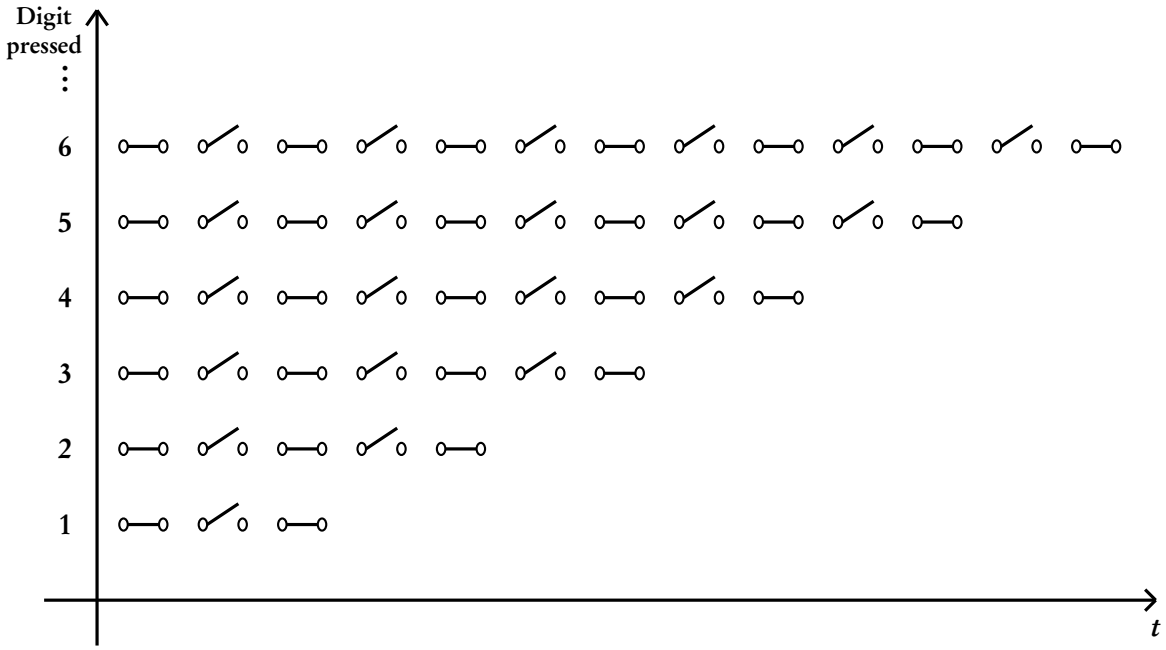

   (d) Explain the operation of an edge-triggered D-type flip flop. Please include a schematic diagram and ensure to include thorough explanations of the setup time, hold time and the delay from a clock edge to the output. What happens if the setup time or hold time are violated? How could such violations occur?

   (e) Suppose I design a new D-type flip flop circuit, which advertises a setup time of zero. Why would it be undesirable to use such a D-type flip flop in a shift register design?

   (f) Explain the operation of a tristate output. How are such outputs used?

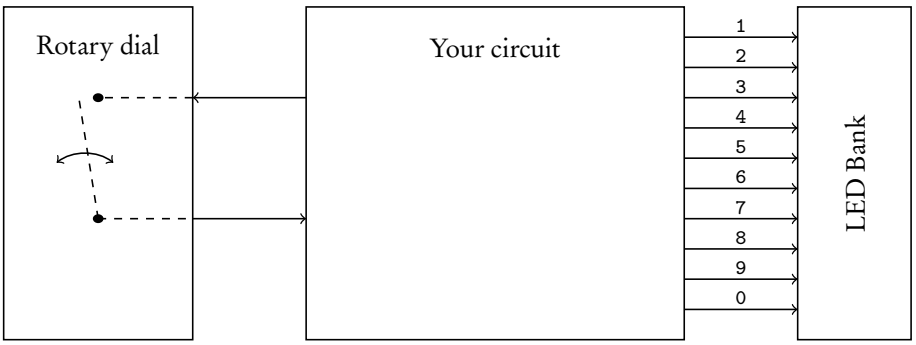
6.

A problem engineers often face is how to enable humans to enter data into the systems that they design. Until DTMF was widely deployed the 1970s, telephone networks primarily used *pulse dialling* for humans to enter the number of the persons to whom they wished to speak. The person making a telephone call would turn a *rotary dial* to enter each digit of the recipient's telephone number. Supposing the person entered the number $n$ ($n \in \{0, \ldots, 9\}$), the rotary dial would correspondingly interrupt a direct current local loop circuit $n$ times. Essentially, the operation can be thought of as opening and closing a switch to generate a train of pulses as follows:

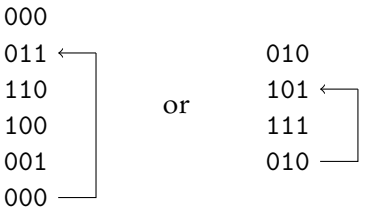



For $x$-axis scale, pulse rates on the edge (consumer) network were—and indeed still are when pulse dial telephones are used—around 10 pulses per second.

Your task is to design a circuit to convert these switch operations into a parallel output, with one wire for each digit. The rotary dial should plug directly in to your circuit. The signal on the wire corresponding to a particular digit should be high if, and only if, that digit was dialled on the rotary dial. The circuit need only deal with a single digit being pressed: there is no need to consider multiple digits, or an entire telephone number. You might like to include a RESET input that can be pressed after a particular input has been decoded, and the circuit is ready to accept the next input.



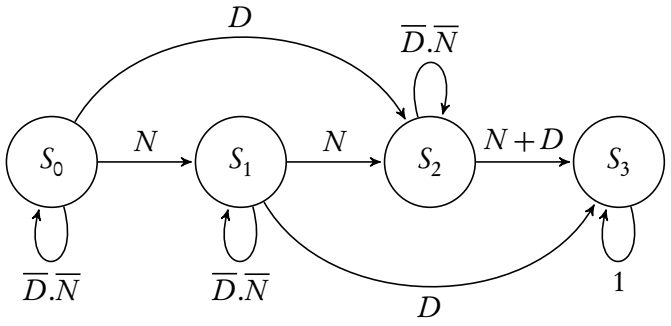

7. [AM] A sequential circuit has been built using D-type flip flops. It behaves slightly erratically. When switched on, it produces on its three output wires one of the following patterns:




Deduce the circuit details by drawing the next-state table and working out a minimised sum-of-products expression for the input to each of the three state registers ($CBA$). Propose a modification that ensures that in due course the circuit will always settle into the cycle shown in the first pattern.

8. [AM] A 3-bit synchronous counter has a mode control input $X$. If $X = 0$, the counter steps through the binary sequence 111, 110, 101, 100, 011, 010, 001, 000, and repeat. Alternatively if $X = 1$, the counter advances through the Grey code sequence 111, 101, 100, 000, 001, 011, 010, 110, and repeat. Draw the state diagram for the counter and deduce the next-state logic, assuming the use of D-type flip flops for the three state registers. Draw the resulting circuit.

9. [AM] A machine has the state diagram shown below, where $N$ and $D$ are the two inputs and $N = D = 1$ cannot occur. The state assignment is $S_0 = [00]$, $S_1 = [01]$, $S_2 = [10]$ and $S_3 = [11]$, where the machine starts in state $S_0$ and finishes in state $S_3$. Note that the state $= [Q_1 Q_0]$ where $Q_n$ is the output of flip-flop $n$.




   (a) Write down the state transition table for this machine.

   (b) Assuming the use of J-K flip-flops for the state registers, write down the modified state transition table and determine the minimised Boolean expressions for the next state functions.

## Extension questions

10. Design an asynchronous circuit with inputs $A$ and $B$ and output $Y$ that has the following truth table:

| $A$ | $B$ | $Y_t$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | $Y_{t-1}$ |
| 1 | 0 | $Y_{t-1}$ |
| 1 | 1 | 1 |

    where $Y_{t-1}$ denotes a *don't change* condition.

    That is, use SR latches and other components to design a circuit that:

    (a) reflects its inputs when the states of all the inputs match, and

    (b) whose output should remain in this state until the inputs all transition to the other state.

    Redesign your circuit using only NAND gates. How small can you make it?

11. Examination question: 1996 Paper 2 Question 3 (attached at the end of this document).

12. A scheme for *carry look-ahead* is described in the lectures to improve the time performance of conventional ripple carry adders. Research other alternative approaches for handling carry in the ALU, and describe:

    - the *Manchester* carry chain
    - carry skip

- carry select

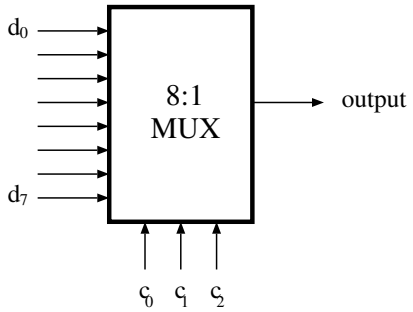
Fill in the following table of their complexities using big-O notation, to summarise their performance:

| | **Time** | **Space** |
|---|---|---|
| Ripple | | |
| Carry look-ahead | | |
| Manchester | | |
| Carry skip | | |
| Carry select | | |

# 1996 Paper 2 Question 3

## Digital Electronics

($a$) A multiplexer is a device that selects one of its inputs as the output. The selection is determined by a set of control signals. For example, in the 8:1 multiplexer shown below, the output will be equal to $d_6$ when $c_2 = 1$, $c_1 = 1$ and $c_0 = 0$.




Give a circuit which implements this 8:1 multiplexer using only NAND gates. [10 marks]

($b$) Using only 8:1 multiplexers, show how to build a 16:1 multiplexer. [4 marks]

($c$) Show how an 8:1 multiplexer and a single inverter can be used to implement any combinational function of four variables. (You may assume the availability of signals for logical 1 and logical 0.) [6 marks]