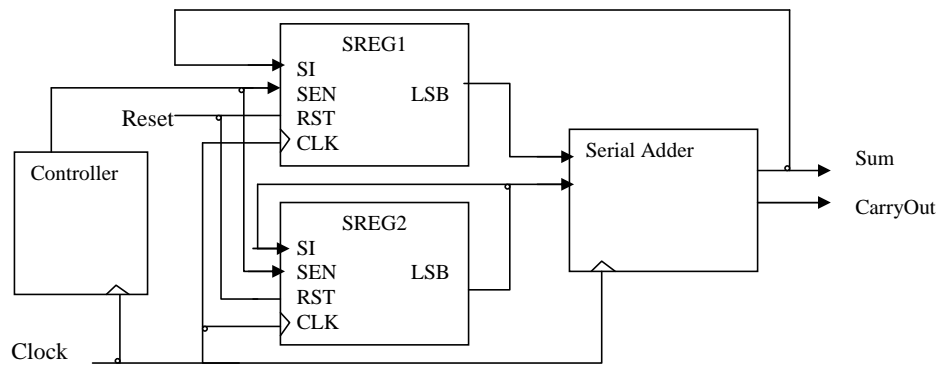


Q.1 (parts a-c on following pages)

Consider the serial adder below. Two shift registers *SREG1* and *SREG2* are used to hold the four bit numbers to be added. Each register has a Shift Enable signal *SEN*, serial input *SI*, Clock and Reset. When *SEN* = 1 and the Clock is pulsed, *SI* is entered into the Most significant bit, MSB, of the register, as the contents of the register are shifted right one position. The connections needed for initial loading of the registers are not shown. The upper register serves as accumulator, and after four shifts its contents is replaced with the sum of the initial numbers. The lower register is connected as a cyclic shift registers, so after four shifts it is back to its original state and the second number is not lost.



- (a) [Design the Serial Adder block for the circuit above. It should include a full adder and a flip-flop to store carry, to be used at the next clock cycle. At each clock cycle, one pair of bits is added in the full adder. It should also include a logic which must accommodate the following operation involving Clock and the *SEN* signal: when *SEN* is 1, the carry bit is stored in the carry flip-flop on the clock edge.

- (b) Design the Controller (based on a Mealy FSM) so that after receiving a start signal, N , it will output $SEN = 1$ for four clocks and then stop. $SEN = 1$ shifts the sum bit into the upper register and causes the lower register to rotate right. Draw the state graph and the state table. Assume that the start signal N is terminated before the network returns to the first state, so no further action occurs until another start signal is received. Once the second state is reached, the network operation continues regardless of the value of N .

- (c) Provide an alternative high-level FSM for the FSM in part (b).

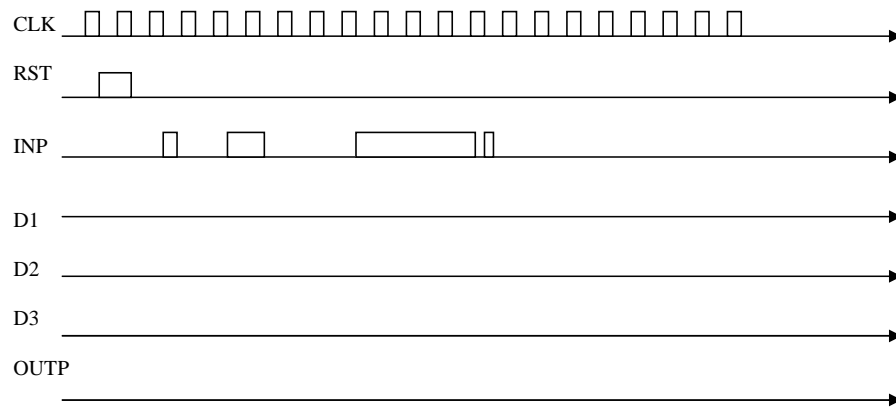
Consider the VHDL code that defines a debounce circuit, which can be used to qualify the input of a push-button switch.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;

entity DEBOUNCE is
    port(INP, RST, CLK: in std_logic;
         OUTP: out std_logic);
end entity DEBOUNCE;

architecture BEHAV of DEBOUNCE is
    signal D1, D2, D3: std_logic;
begin
    process (CLK, RST)
    begin
        if (RST = '1') then
            D1 <= '0';
            D2 <= '0';
            D3 <= '0';
        elsif (CLK'event and CLK = '1') then
            D1 <= INP;
            D2 <= D1;
            D3 <= D2;);
        end if;
    end process;
    OUTP <= D1 and D2 and (not D3);
end architecture BEHAV;
```

- (a) Draw the circuit and its timing diagram (see the template below) to answer the question: why this circuit is called a debounce circuit?



- (b) In addition, the circuit ensures that the output indicating the button press is of fixed duration regardless of how long the actual button is pressed. How long is this fixed duration?

- (c) Propose an alternative debounce circuit solution, based on a counter (draw the design schematics). Specify the counter size.

- (d) Provide the VHDL fragment corresponding to the debouncer based on the counter from part (c) in the space provided below.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;

entity DEBOUNCE1 is
    port

end entity DEBOUNCE1;
architecture BEHAV of DEBOUNCE1 is

end architecture BEHAV;
```

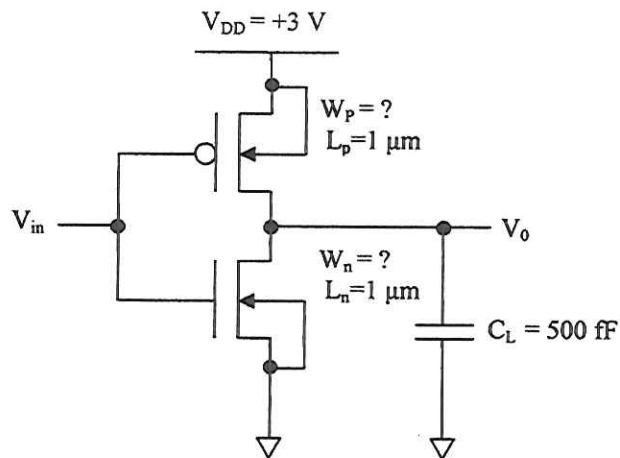
Q.1 Design a static CMOS inverter (see figure 2), using $1\mu\text{m}$ long nMOS and pMOS transistors and of appropriate widths, so as to meet the following performance requirements:

- 1) A falling delay of 0.2 ns , for an output transition from 2 V to 0.6 V , assuming an output load capacitance of 500 fF and ideal step input of 3 V being applied to the inverter.
- 2) An inverter switching threshold voltage $V_{INV} = 1.4\text{ V}$.

$2\mu\text{m}$ CMOS device and process parameters

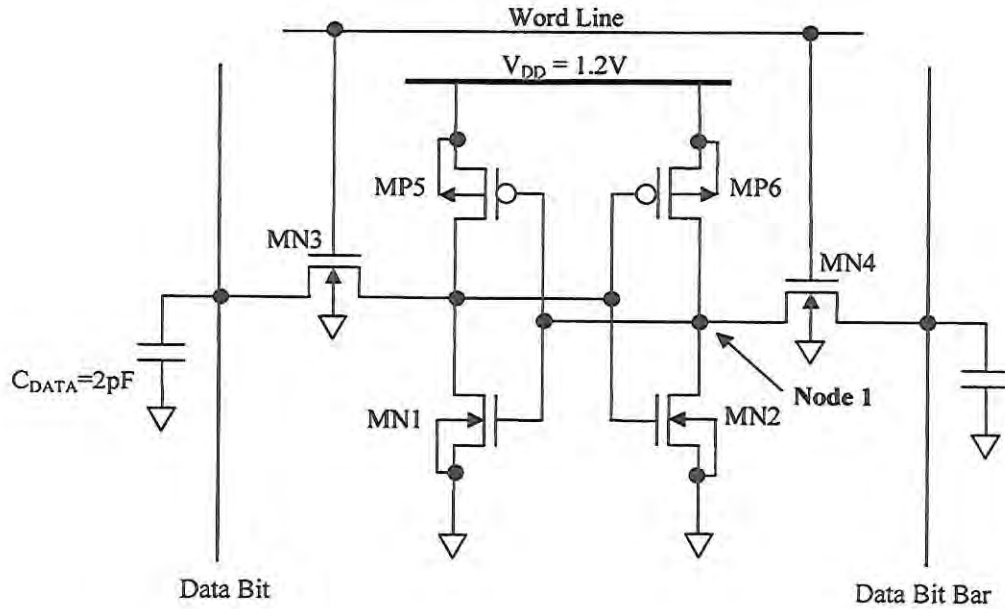
$$k'_n = 100 \mu\text{A}\cdot\text{V}^{-2}; V_{T0(n)} = 0.8\text{ V}; L_n = 1\mu\text{m}; W_{n(\text{min})} = 2\mu\text{m}; \gamma_n = 0.4\text{ V}^{0.5}; |2\phi_{F(n)}| = 0.6\text{ V}$$

$$k'_p = 50 \mu\text{A}\cdot\text{V}^{-2}; V_{T0(p)} = -1.0\text{ V}; L_p = 1\mu\text{m}; W_{p(\text{min})} = 2\mu\text{m}; \gamma_p = 0.4\text{ V}^{0.5}; |2\phi_{F(p)}| = 0.6\text{ V}$$



Q.2 What are the respective widths of transistors MN4 and MP6 of the CMOS static memory cell shown below, so that the design meets the following requirements:

(a) The voltage at **node 1** during the **WRITE '1'** operation must be pulled down to at least **0.4 V**.

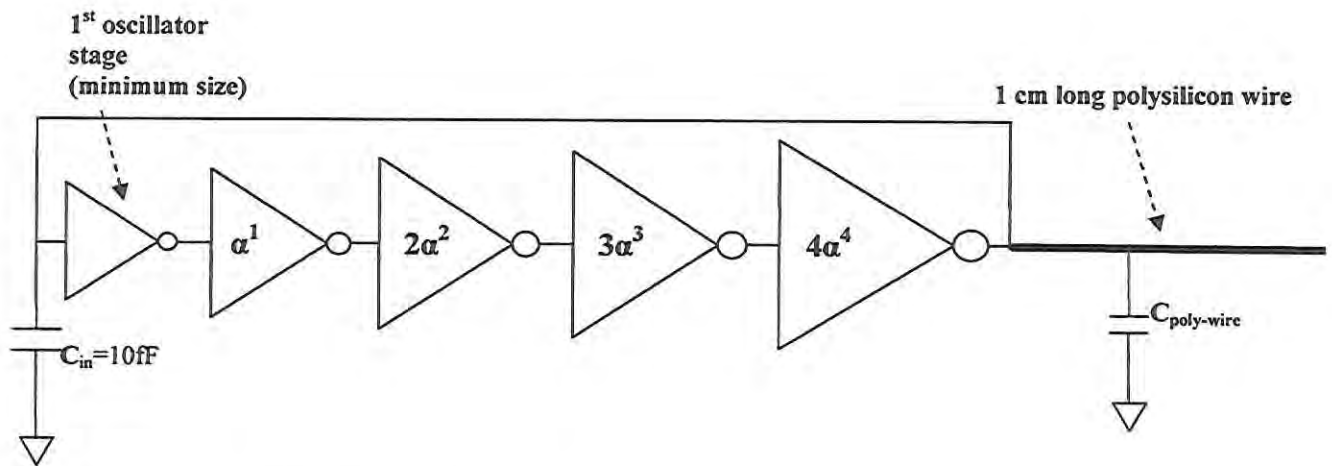


$$k'_p = 130 \mu\text{A} \cdot \text{V}^{-2}; k'_n = 400 \mu\text{A} \cdot \text{V}^{-2}; V_{T0(n)} = 0.4 \text{ V}; V_{T0(p)} = -0.4 \text{ V}; \gamma_n = 0.2 \text{ V}^{1/2}; \gamma_p = 0.2 \text{ V}^{1/2}$$

$$|2\phi_F| = 0.88 \text{ V}; L_{MN1} = L_{MN2} = L_{MN3} = L_{MN4} = L_{MP5} = L_{MP6} = 0.1 \mu\text{m};$$

- Q.3 In order to generate an on-chip clock, a tapered five stage ring oscillator is used in order to drive a long capacitive polysilicon wire, as shown below. The first ring oscillator stage is a minimum sized inverter with $C_{in} = 10\text{fF}$, and a propagation delay of 175 pS , when driving a similar minimum sized inverter. The sizes of the 2nd and subsequent stages are scaled by α , $2\alpha^2$, $3\alpha^3$, $4\alpha^4$ respectively. Given that that polysilicon line width is $5\mu\text{m}$ with a per unit area capacitance of $0.05\text{ fF}/\mu\text{m}^2$
- Determine the optimum scaling factor (α) of the ring oscillator stages in order minimize the propagation delay.

- What is the frequency of oscillation of the tapered ring oscillator?



A 5 stage tapered ring oscillator driving a polysilicon wire

TOPIC: Embedded system interfacing (ENCM511)

- 1) A developer has a microprocessor that transmits to 4 external peripherals over 3 SPI lines, a common industrial situation.
 - (i) Draw a labelled diagram of this configuration.
 - (ii) Using this diagram, explain the basics of the SPI interface, and detail how it is possible for a microprocessor to control 4 peripherals over 3 SPI lines
 - (iii) Actual there are 5 spi-lines needed if you take into account the transmission of information from the peripherals to the processor – MISO and GND (ground). Explain why the performance of the SPI interface will / may be unreliable if there is a missing ground wire
- 2) Using brief examples, explain the relationship between and / or differences off these terms associated with editing a program: compiling, linking and downloading the code to a microprocessor evaluation board.
Source file, Object files, Executable, library file, linker error, JTAG- interface, boot-code
- 3) Discuss and calculate the execution time difference of C++ algorithm running on a 300 MHz processor designed with and without a 4 instruction pipeline (Fetch, decode, execute, writeback). Assume that the machine code generated from the C++ algorithm requires 1 jump / call / return instruction for every 20 lines of code.
- 4) Discuss and calculate the execution time difference of C++ algorithm running on a 300 MHz processor designed with and without a 4 instruction pipeline (Fetch, decode, execute, writeback) when an external device is generating a 1 MHz interrupt signal which are serviced by a short ISR routine (100 instructions). Assume that the machine code generated from the C++ algorithm requires 1 jump / call / return instruction for every 20 lines of code.

TOPIC: Advanced processor architectures (ENCM515)

- 1) Draw labelled diagrams that illustrate the basic characteristics of Harvard and von-Neumann processors. Explain how these architectural differences impacts the execution times of the following 2 functions.

<pre>float DoSum(int number) { float sum = 0.0; for (int count = 0; count < number) { sum = 2 * count; } return sum; }</pre>	<pre>float DoSumWithMemory(int number, float *pt){ float sum = 0.0; for (int count = 0; count < number) { sum = 2 * count + *pt++; } return sum; }</pre>
---	---

- 2) Explain and contrast the execution speed advantages of the following concepts found on many modern processor:
 - a. SISD, SIMD, super-scalar instructions
 - b. Large data blocks (256 WORDS OR MORE) being send to peripherals controlled with, or without, the use of DMA activity
- 3) A key advantage of many modern microprocessors is their ability to handle real-time audio operation, e.g. digital filtering.
 - a. Explain what is meant by “real-time” or “non-real” time audio operations
 - b. Draw the block diagram of a microprocessor system with A/D and D/A capable of sampling at 8 kHz, which has been coded with a low-pass digital filter with 400 Hz bandwidth.
 - c. For your system, what will be the output from the processor if it is receiving an audio signal with components at 65Hz, 2065Hz and 4065Hz? Explain your answer.

TOPIC: Computer Arithmetic & Computational Complexity (ENCM517)

Q1. Convert integers (say, 1009) from binary to Booth encoding and explain why it is beneficial for implementing a constant multiplication operation

Q2. Show the work of RSA encryption algorithm for a specified key parameters.

Q3. Compute the connectivity matrix of a given graph

Q4. Show the relationship between given algorithmic problems (say, solving system of linear equations for a) symmetric and b) non-symmetric matrices).