Assignment 1

ELEC 403

Isabelle André – 12521589

1.0 TX FIR Response & Nyquist Frequency

Figure 1 shows the TX FIR frequency response along with the channel and combined frequency response.

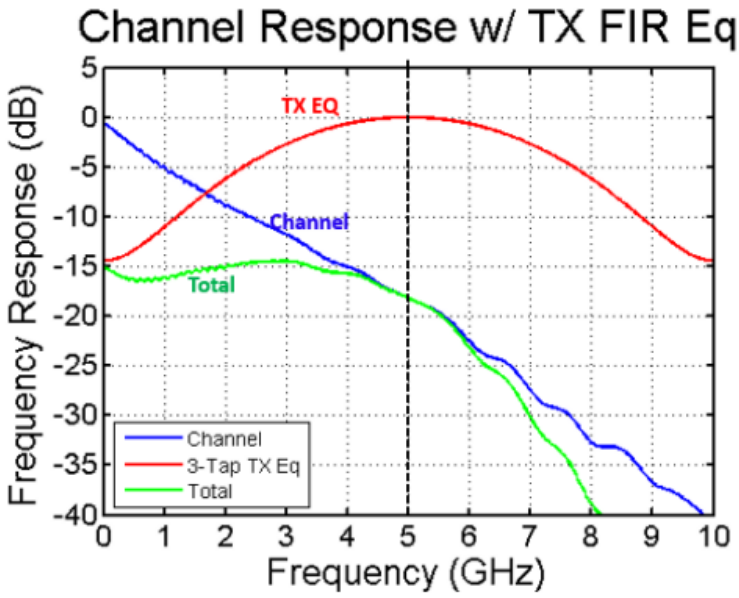

Figure 1: Channel Response with TX FIR Equalization

The 3-Tap TX EQ response is symmetrical around the Nyquist frequency. The Nyquist frequency is defined as half the sampling rate of a discrete-time system, and the upper limit of the frequency range that can be accurately represented in the discrete domain without aliasing. A 3-Tap FIR Equalizer can be described as:

$$W(z) = w[-1] + w[0]z^{-1} + w[1]z^{-2}$$

where w are the tap coefficients. After converting into the z-domain, the following periodic response is obtained:

$$z = e^{sTs} = e^{j2\pi fTs} = cos(2\pi fT_s) + jsin(2\pi fT_s)$$

where Ts is the period. It is observed that the value of z is equivalent at every interval of the Nyquist frequency $f_N = \frac{1}{2Ts}$. In the frequency domain, this same value of z will result in the same value of W(z). As the signal is periodic in the frequency domain, the 3-Tap TX EQ response will be symmetrical around any multiple of the Nyquist Frequency.

2.0 Linear Feedback Shift Register (LFSR) for PRBS7 Generator

Figure 2 shows a linear feedback shift register (LFSR) circuit for a PRBS7 generator, with a polynomial given by $x^7 + x^6 + 1$.

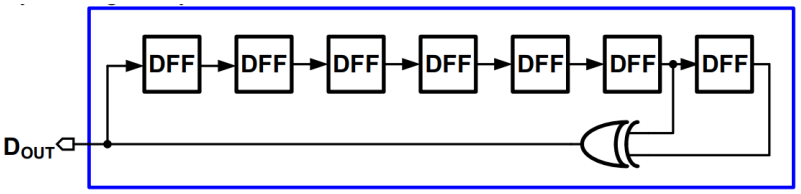

Figure 2: Linear Feedback Shift Register (LFSR) for PRBS7 Generator

2.1 PRBS3 Generator

Figure 3 shows the corresponding circuit for a PRBS3 generator, with a polynomial given by $x^3 + x^2 + 1$.

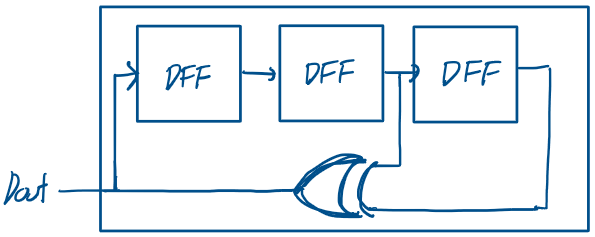

Figure 3: PRBS3 Generator

2.2 30 Cycles Bit Sequence

The output of the 1st DFF is 1, and all other DFFs have 0 output when the circuit is turned from reset. At reset, the sequence 100 is stored. Each bit are shifted down at every clock cycle. The Bit sequence for Dout is shown below for 30 clock cycles.

Cycle	0	1	2	3	4	5	6	7	8	9	10
Dout	0	1	1	1	0	0	1	0	1	1	1

Cycle	11	12	13	14	15	16	17	18	19	20
Dout	0	0	1	0	1	1	1	0	0	1

Cycle	21	22	23	24	25	26	27	28	29	30
Dout	0	1	1	1	0	0	1	0	1	1

2.3 PRBS3 Sequence Repetition

According to the bit sequences in Section 2.2, the maximum number of bits in the PRBS sequence before being repeated is 7 bits, for example the output pattern from cycles 0-6 repeats itself from cycles 7-13.

2.4 Maximum 1 Bit Run Length

The maximum run length for 1 bit is 3, for example from cycle 1-3.

2.5 Unique Non-Zero 3-Bit Binary Words

Grouping every 3-bit sequence in the PRBS3 sequence and wrapping around to the beginning for the final 3-bit sequence, there are 7 unique non-zero 3-bit words.

1	2	3	4	5	6	7	8	9	10
011	**100**	**101**	**110**	**010**	**111**	**001**	011	100	101

2.6 PRBS-k Sequence

- For a PRBS-k sequence, the sequence repeats every $2^k - 1$ cycles, where k is the number of DFFs. There are 2^k possible combinations of bits for a PRBS-k generator, however as the sequence can never output k times 0 bits due to the XOR, we subtract 1 for $2^k - 1$ bits.
- The maximum 1 bit run length is k due to the XOR, as a 0 will be outputted next once a sequence of 3 1 bits is produced for a PRBS-3.
- Once again, there are $2^k - 1$ unique non-zero k-bit words, as there are $2^k - 1$ possibilities for k number of bits, without counting a sequence of zeros.

3.0 8T Dual-Port SRAM Cell

A standard single-port 6T SRAM cell consists of two pull-up PMOS and two pull-down NMOS as well as two NMOS connecting the cross-coupled inverters to the two complimentary bitlines as shown in Figure 4. As this cell has only one word line and one pair of complementary bitlines, this cell can only be used in a single read or write operation at once, therefore 1WR.

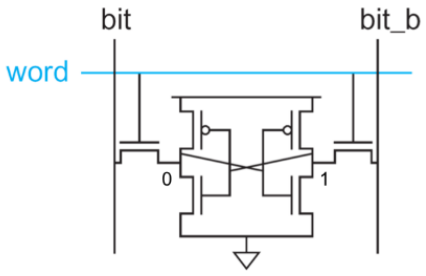

Figure 4: 6T Single-Port SRAM Cell

Figure 5 shows an 8T-SRAM cell in a 1W1R configuration. This is a dual-port storage cell, as indicated by its two sets of address and bit lines. One port is used exclusively for write operations while the other is used for read operations.

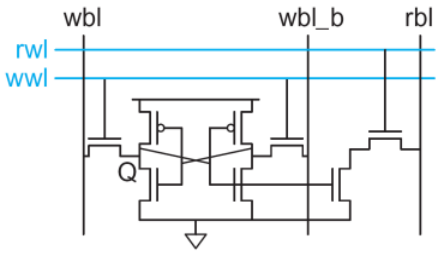

Figure 5: 8T SRAM Cell (1W1R)

The write operation functions in a similar manner as a 6T SRAM cell. When the write word line (wwl) is active, write bitline (wbl) and its complimentary signal write bitline bar (wbl_b) pull the data to store in the cell.

The read operation uses a different data path than the write. Assuming that Q stores bit 0, then Q_bar on the opposite inverter stores bit 1. When the read word line (rwl) is active, if Q stores a 0, then it will be transmitted to the read bitline (rbl) output. If Q stores a 1, an outside circuit is needed to produce a 1 similarly to Figure 6, as there is no path from ground to the read output. The external circuit is a pull-up circuit to generate a 1. At the rising edge of the clock, if the bit stored is a 1, the read bitline disconnects, therefore conducting the external circuit's default 1 value as the output, otherwise if a 0 is stored, the read data path is connected to ground, outputting a 0. An RS flip flop is used to maintain the value being read.

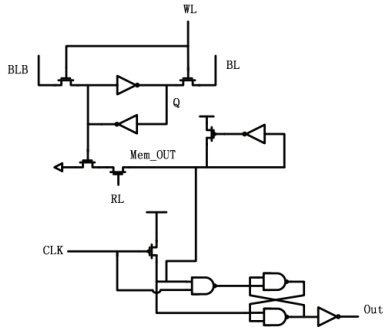

Figure 6: 1W1R Cell with External Circuit

While two extra transistors increases the size of the cell by 20%, consumes more power, and adds complexity compared to a 6T SRAM cell, other benefits may outweigh this drawback. For instance, read operations do not affect the contents of the cell as the worst-case static noise margin is only that of the cross-coupled inverters. This results in a much better static noise margin than the 6T SRAM.

8T SRAM cells are designed to be more stable and less prone to data loss or corruption. Using a separate read and write system, the frequency of the system can be increased by writing a bit at the negative edge of the clock, then reading it at the positive edge of the clock. This gives a better read and write performance than a standard 6T SRAM cell as read operations do not disturb the stored data. This separate read channel also does not need to be pre-charged before a read is initiated, and the read channel does not demand the size of the transistors by CR requirements.

Source: [A research of dual-port SRAM cell using 8T | IEEE Conference Publication | IEEE Xplore]

4.0 2-Input AND Gate Design

Figure 7 shows two different configurations of an AND gate.

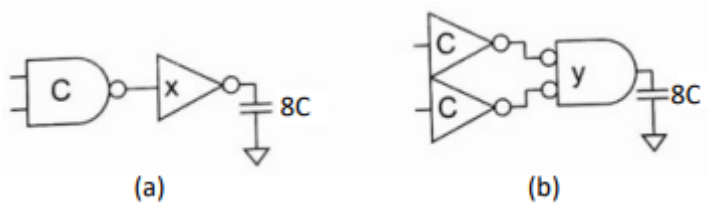

Figure 7: 2-Input AND Gate

4.1 Intuition

PMOS transistors have less mobility than NMOS due to its W/L ratio and higher impedance. As shown in Figure 8, the NAND2 has 2 PMOS in parallel, NOR2 gates have two PMOS connected in series to pull the output high and have a greater area than NAND2 gates, resulting in a path with higher impedance. Therefore, NAND2 gates are generally faster than NOR2.

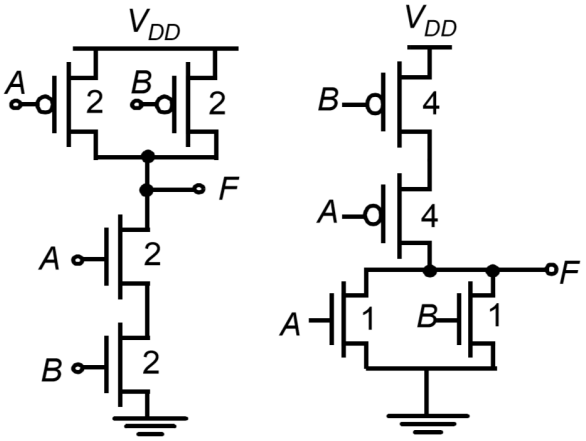

Figure 8: NAND2 (left) and NOR2 (right) Transistors

4.2 Path Effort

The path effort F for each design can be computed as follows:

$$F = GBH$$

where G is the logical effort, B is the branching effort, and H is the electrical effort. The logical effort G for a path is the product of every stage's logical effort. The logical effort of common gates are shown in Table 1. For an inverter, g = 1. for a 2 input NAND2 gate, g = 4/3, and for a 2 input NOR2 gate, g = 5/3.

Therefore, the logical effort G for design a) can be calculated as the product of each stage, $G = 4/3 * 1 = 4/3$. For design b), $G = 1 * 5/3 = 5/3$.

The branching effort B is the product of branching between stages in path, defined as:

$$b = \frac{C_{on\ path} + C_{off\ path}}{C_{on\ path}}$$

As there is no fanout or branching along either designs, the branching effort is B = 1.

Finally, the electrical effort H is the fanout or ratio of output to input capacitance:

$$H = \frac{C_{out-path}}{C_{in-path}}$$

For design a) and b), the electrical effort is $H = 8C/C = 8$.

Therefore, for design a), the total path effort is $F = 4/3 * 1 * 8 = 32/3$ and for design b), $F = 5/3 * 1 * 8 = 40/3$.

4.3 Delay

The delay of the design is defined as the sum of all stage delays or the sum of the path effort delay and parasitic delay:

$$D = \sum d = D_F + P = NF^{1/N} + P$$

where N is the number of stages N = 2. The effort delay is the sum of the logical effort and electrical effort $f = gh$. The parasitic delay represents the delay of the gate at no load due to internal parasitic capacitance. Using the values found in section 4.2, for design a), the path effort delay is $D_F = 2 * (\frac{32}{3})^{1/2}$, and the parasitic delay is $P = 2 + 1 = 3$, therefore $D = 9.532$.

For design b), $D_F = 2 * (\frac{40}{3})^{1/2}$ and the parasitic delay is $P = 1 + 2 = 3$. Therefore, the path delay is $D = 10.303$.

4.4 Input Capacitance

To calculate the input capacitances x and y required to achieve the path delay in Section 4.3:

$$C_{in} = \frac{gC_{out}}{\hat{f}}$$

where g is the logical effort, and $\hat{f}$ is the best stage effort, $\hat{f} = F^{1/N}$. From Section 4.2, F = 32/2, N = 2, and g = 1 for the inverter in design a). The input capacitance x of the inverter can be found by setting finding $\hat{f} = (\frac{32}{3})^{1/2} = 3.266$. Then $x = C_{in} = \frac{1*8C}{3.266} = 2.449C$.

For design b), F = 40/3, N = 2, and g = 5/3 for the NOR gate. $\hat{f} = F^{1/N} = (\frac{40}{3})^{1/2} = 3.651$. Therefore, for the inverter's input capacitance y: $y = C_{in} = \frac{5/3 * 8C}{3.651} = 3.652C$.

5.0 Logical Effort of N-Input NAND and N-Input NOR

5.1 Logical Effort of N-Input NAND gate

Figure 9 shows a single inverter with the PMOS sized k times the size of the NMOS, resulting in k times the effective resistance.

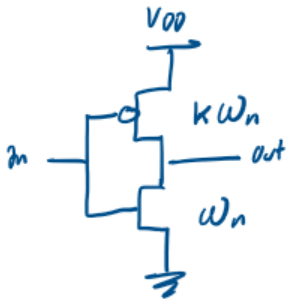

Figure 9: Inverter With PMOS Sized k Times NMOS Size

An N-Input NAND Gate is shown in Figure 10. As NMOS are stacked in series while PMOS are in parallel, the sizing for the NMOS increases with the number of inputs, N*Wn while the sizing for PMOS remains k*Wn.

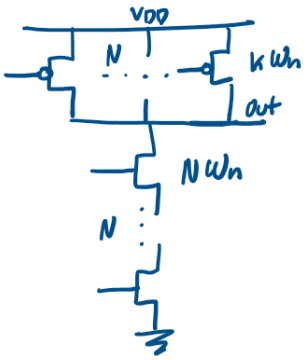

Figure 10: N-Input NAND Gate

A general expression for logical efforts of a N-input NAND gate can be estimated by counting transistor widths and finding the ratio of input capacitance of a gate to the input capacitance of an inverter delivering the same output current. For an inverter, the input capacitance is $C_{INV} = kWn + Wn = Wn\,(k + 1)$. For the N input NAND gate, the input capacitance is $C_{NAND} = NWn + kWn = Wn(N + k)$. Therefore, the logic effort for an N-input NAND gate can be approximated to $g = \frac{C_{NAND}}{C_{INV}} = \frac{N+k}{k+1}$.

5.2 Logical Effort of N-Input NAND gate

An N-Input NAND Gate is shown in Figure 11. As NMOS are in parallel while PMOS are in series, the sizing for the PMOS increases with the number of inputs, N*k*Wn while the sizing for NMOS remains Wn.

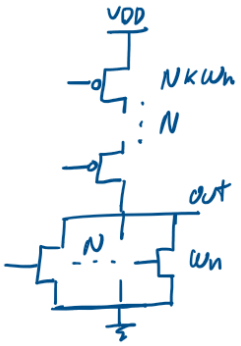

Figure 11: N-Input NOR Gate

The input capacitance for the NOR gate is $C_{NOR} = NkWn + Wn = Wn(kN + 1)$. Therefore, the logic effort for an N-input NOR gate can be approximated to $g = \frac{C_{NOR}}{C_{INV}} = \frac{kN+1}{k+1}$.

5.3 NAND and NOR Comparison

The logical effort for a NOR gate is greater than the logical effort for a NAND gate when k>1. NAND gates scale better with the number of inputs than a NOR gate, as the PMOS are in series in a NOR gate, resulting in a multiple of kN in the input capacitance. The logical effort increases faster with the number of inputs. Therefore, a NAND-N gate would be more desirable than a NOR-N gate in the regular case of k>1.

6.0 Inverters in Output Bump Pad

An output bump pad contains a chain of successively larger inverters to drive an off-chip load capacitance of 200fF as shown in Figure 12. The first inverter in the chain has an input capacitance of 2fF.

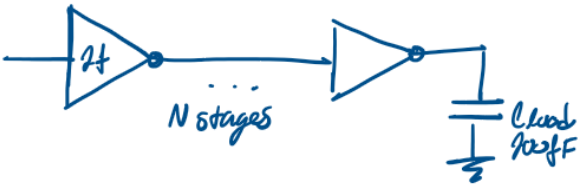

Figure 12: N stage Inverter Output Bump Pad

6.1 Inverters Driving Load

The best number of stages is a fanout of 4 (FO4) given by $N = log_4(F)$, where F is the path effort, F = BGH. B is the branching effort, B = 1 for a single branch. G is the logical effort defined as $G = \prod g_i = 1$ for an N stage chain of inverters, and H is the electrical effort, defined as the ratio of output to input capacitance, $H = \frac{C_{out}}{C_{in}} = \frac{200}{2} = 100$. The path effort $F = 1 * 1 * 100 = 100$, and $N = log_4(100) = 3.322$. Therefore, a 3-stage inverter chain design is used to drive the load with least delay.

6.2 Delay Estimation in FO4 Inverter Delays

The least delay for the 3-stage inverter chain expressed in FO4 delays can be found as $D = NF^{1/N} + P$, where N = 3, F = 100, and the parasitic delay $P = \sum p_i = 3$ for a 3-stage inverter. Therefore, $D = 3 * (100)^{1/3} + 3 = 16.925$.

7.0 1-2 Fork Clock Buffer

Figure 13 shows an example of a 1-2 fork clock buffer. Both true and complementary outputs drive a load of C2. The 1-2 fork has a maximum input capacitance of C1.

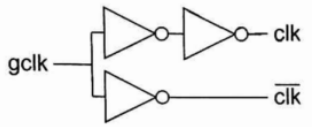

Figure 13: 1-2 Fork Clock Buffer

7.1 Inverter Capacitance

To find the capacitance of each inverter, we assign an input capacitance of x to the top branch of the fork, and C1-x to the bottom branch, to consider the two branches separately as shown in Figure 14. The delay of the two branches must match, therefore inverters must be sized accordingly.

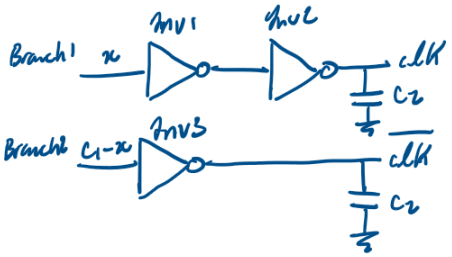

Figure 14: 1-2 Fork Clock Buffer Branch Split

For Branch 1, the input capacitance of INV1 is set to x. To find the input capacitance of INV2, we use:

$$C_{in} = \frac{gC_{out}}{\hat{f}}$$

where the best stage effort $\hat{f} = F^{1/N}$ and N=2 for two stages. The effort delay F is F=GBH, where the branching effort B=1 for one branch, the logical effort G=1 for inverters, and the electrical effort $H = \frac{C_{out}}{C_{in}} = \frac{C_2}{x}$. Therefore, the effort delay is $F = 1 * 1 * \frac{C_2}{x} = \frac{C_2}{x}$, the best stage delay is $\hat{f} = (\frac{C_2}{x})^{1/2}$. The input capacitance of INV2 is $Cin = \frac{1*C2}{(\frac{C2}{x})^{1/2}} = (xC_2)^{1/2}$.

For branch 2, the input capacitance of INV3 is simply C1-x.

7.2 Circuit Operation Speed

The circuit operation speed in terms of p_{inv} can be found using the least delay $D = NF^{1/N} + P$ where for branch 1, N=2 stages, $F^{1/N} = (\frac{C_2}{x})^{1/2}$ from Section 7.1, and the parasitic delay $P = \sum p = 2p_{inv}$ for 2 inverters. Therefore, the delay for branch 1 is $D_1 = 2(\frac{C_2}{x})^{1/2} + 2p_{inv}$.

For branch 2, N=1 stages, F=GBH where G=1, B=1, and $H = \frac{Cout}{Cin} = \frac{C_2}{C_1 - x}$. The effort delay is $F = 1 * 1 * \frac{C_2}{C_1 - x}$, th parasitic delay is $P = \sum p = p_{inv}$ for 1 inverter, therefore the delay is $D_2 = \frac{C_2}{C_1 - x} + p_{inv}$.

As the delay of the two branches must match, we can set $D_1 = D_2$ and $2(\frac{C_2}{x})^{1/2} + 2p_{inv} = \frac{C_2}{C_1 - x} + p_{inv}$. By setting the capacitances for C1 and C2, we can solve for x at different values of p_{inv} then solve for D1 or D2 to find the circuit's operation speed.