Kyle Loyka

Homework 4

ECEN 449-503

Due: 26 April 2016

Kyle Layka
ECEN 499

Assignment 4

1. Transfer Rate: 30 Gb/s $\rightarrow \frac{1 \text{ sec}}{30 \text{ Gb}} = 0.0333 \text{ ns}$ per bit.

a) PPM @ 16 values $\Rightarrow$ need 16 bits for 1 sample

$(0.0333 \text{ ns})(16) = 0.5333 \text{ ns}$ per sample

- full duplex means two samples sent (per sample period) for a phone call $\Rightarrow$ 1.066666 ns per sample
- Nyquist: 2B sample rate $\rightarrow$ 2(16 KHz) = 32 KHz $\rightarrow$ 32000 samples

32000 · 1.066666 ns = 34.133 μs = time to transfer both samples out of 1 second

- $\frac{1}{34.133 \mu s} \approx$ **29296 calls**

b) PCM @ 16 values $\Rightarrow$ need 4 bits for 1 sample

$(0.0333 \text{ ns})(4) = 0.1333 \text{ ns}$ per sample

- full duplex $\Rightarrow$ 1 call :: 0.266666 ns
- Nyquist: (32000 samples)(0.266666 ns) = 8.533 μs out of 1 sec
- $\frac{1}{8.533 \mu s} =$ **117192 calls**

2. $f_{frame} = \frac{(32 \cdot 64 + 16) \text{ bits}}{} = 2064 \text{ bits}$. $\frac{40 \text{ KHz}}{\text{bit}} = 82.56 \text{ KHz}$

need a clock rate of 82.56 KHz, with data (a bit) written on the positive edge of clk

compile instructions: using

vcs –gui loyka_kyle_HW4_q3_tb.v loyka_kyle_HW4_q3.v +v2k –sverilog

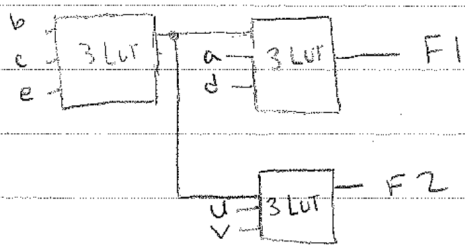
```
// Kyle Loyka
// ECEN 449
// Assigment 3 – Question 3

module PWM_Converter(
  input PWMSIG,
  input ClkSlowPWM,
  input ClkFast,
  output reg PPMSIG,
  output reg ClkSlowPPM
  );

reg delay;
reg flag = 0;

always@(posedge ClkFast) begin
  delay <= ClkSlowPWM;
  ClkSlowPPM <= delay;
end

always@(posedge ClkFast) begin
  if(PWMSIG) flag <= 1;
  else if (!PWMSIG & flag) begin
    flag <= 0;
    PPMSIG <= 1;
  end
  else if(!PWMSIG & !flag) begin
    PPMSIG <= 0;
  end
end
endmodule
```

```
// Kyle Loyka
// ECEN 449
// Assignment 3 – Question 3
// TEST BENCH
`timescale 1ns/1ps

module PWM_Converter_tb;

      reg PWMSIG = 0;
      reg ClkSlowPWM = 0;
      reg ClkFast;
      reg PPMSIG;
      reg ClkSlowPPM;

      /* instantiate unit under test */
      PWM_Converter q3(
            .PWMSIG(PWMSIG),
            .ClkSlowPWM(ClkSlowPWM),
            .ClkFast(ClkFast),
```

```
            .PPMSIG(PPMSIG),
            .ClkSlowPPM(ClkSlowPPM)
      );

      /* setting up clocks */
      initial begin
            ClkFast = 1;
            repeat(325)
             #1 ClkFast = ~ClkFast;
      end

      initial begin
            ClkSlowPWM = 0;
            #16
            repeat(18)
            #16 ClkSlowPWM <= ~ClkSlowPWM;
      end

      /* time to drive signals to the module and see the output */
      initial begin

            int sigs[8] = '{1,2,7,8,10,12,15,3};
            #32 PWMSIG = 0;
            for(int i = 0; i < 8; i++) begin
                  PWMSIG <= 1;
                  #(2*sigs[i]) PWMSIG <= 0; // signals are in units of 2ns
                  #(32-(2*sigs[i])) PWMSIG <= 0; //wait the rest of the time
            end
      end

endmodule
```


DVE - TopLevel.2 - [Wave.1]
File Edit View Simulator Signal Scope Trace Window Help
325000 x1 ps
Any Edge
Name
Value
q3
ClkSlowPWM
St0
PWMSIG
St0
ClkSlowPPM
St0
PPMSIG
St0
ClkFast
St1->St0
Group1
delay
St0
flag
St0
New Group
0
50000
100000
150000
200000
250000
300000
Wave.1
finished: PWM_Converter_tb.unnamed$$_2
325000ps

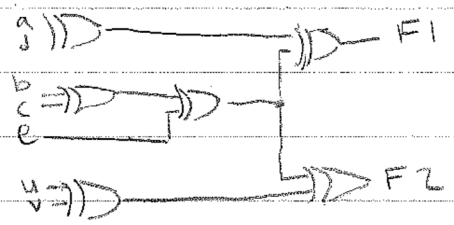
4. a)



F1 a b c d e

F2 b c e u v

b)



(Assume you only have 2-input XOR gates)

MAXIMUM GATE DELAY: 60 ps

MAXIMUM LUT DELAY: 100 ps

5. a) i worst case, polling condition turns true immediately after the polling cycle finished: 600 + 900 [processing] process is already queued

1500 cycles $\left(\frac{1}{1\text{GHz}}\right)$ = 1500 cycles $\left(\frac{1\text{ns}}{\text{cycle}}\right)$ = 1.5 μs

i YES

ii 800 [context switch] + 900 [proc #] = 1700 cycles → 1.7 μs

ii NO

b.) i 600 + 900 = 1500 cycles → 1.5 μs

i YES

ii 800 + 900 = 1700 cycles → 1.7 μs

ii YES