Digitaltechnik
Wintersemester 2021/2022
11. Übung

TECHNISCHE UNIVERSITÄT DARMSTADT

Prof. Dr.-Ing. Thomas Schneider, M.Sc. Daniel Günther, M.Sc. Amos Treiber
Tablet Version **KW04**

Bitte bearbeiten Sie die Übungsblätter bereits im Voraus, sodass Sie Ihre Lösungen zusammen mit Ihren Kommilitonen und Tutoren während der wöchentlichen Übungsstunde diskutieren können.
Mit der angegebenen Bearbeitungszeit für die einzelnen Aufgaben können Sie Ihren Leistungsstand besser einschätzen.

Übung 11.1 Pipelining – Register Transfer Logik [20 min]

Folgende Grafik zeigt eine Pipeline zur Berechnung der Funktion $(x^2+5)\cdot 2-8$:

Mittels eines Eingangsignals **set** wird signalisiert, dass mit einer steigenden Taktflanke ein neuer Wert in das erste Register der Pipeline geladen werden soll. Der in R5 gespeicherte Wert entspricht dem Ergebnis der Funktion.

a) Setzen Sie die Pipeline in SystemVerilog um. Erweitern Sie dafür den unten gegebenen Quelltext für das Pipeline-Modul und entwerfen Sie zusätzliche Module zum Berechnen der Funktion jeder Pipeline-Stufe.

Das Verwenden der arithmetischen Operatoren von SystemVerilog ist in den funktionalen Zusatzmodulen erlaubt. Die Setup/Hold-Zeiten der Register sowie Überläufe können vernachlässigt werden.

seq/pipeline/pipeline.sv

```
1  module pipeline (input logic clock, set, input logic [7:0] in,
2                   output logic [7:0] out);
3
4    // Register zum puffern der Werte zwischen den Pipeline-Stufen
5    logic [7:0] register1, register2, register3, register4, register5;
6
7    // Ausgänge der einzelnen Rechenoperationen
8    logic [7:0] out1, out2 , out3 , out4;
```

b) Erstellen Sie eine Testbench, um die Funktion der Pipeline anhand von einigen Werten per Simulation zu testen.

c) Nehmen Sie nun an, dass die t_{pcq}- sowie t_{setup}-Zeit der Register bei 0,5 ns liegt und die kombinatorischen Schaltungen für die Rechnungen zwischen den Registern kritische Pfade mit folgenden Verzögerungen haben: x^2: 0,6 ns, $+5$: 0,8 ns, $\times 2$: 0,5 ns, -8: 1 ns. Mit welcher Taktfrequenz lässt sich die Pipeline maximal betreiben? Welche Frequenz wäre möglich, wenn man auf Register 2 und 4 verzichtet?

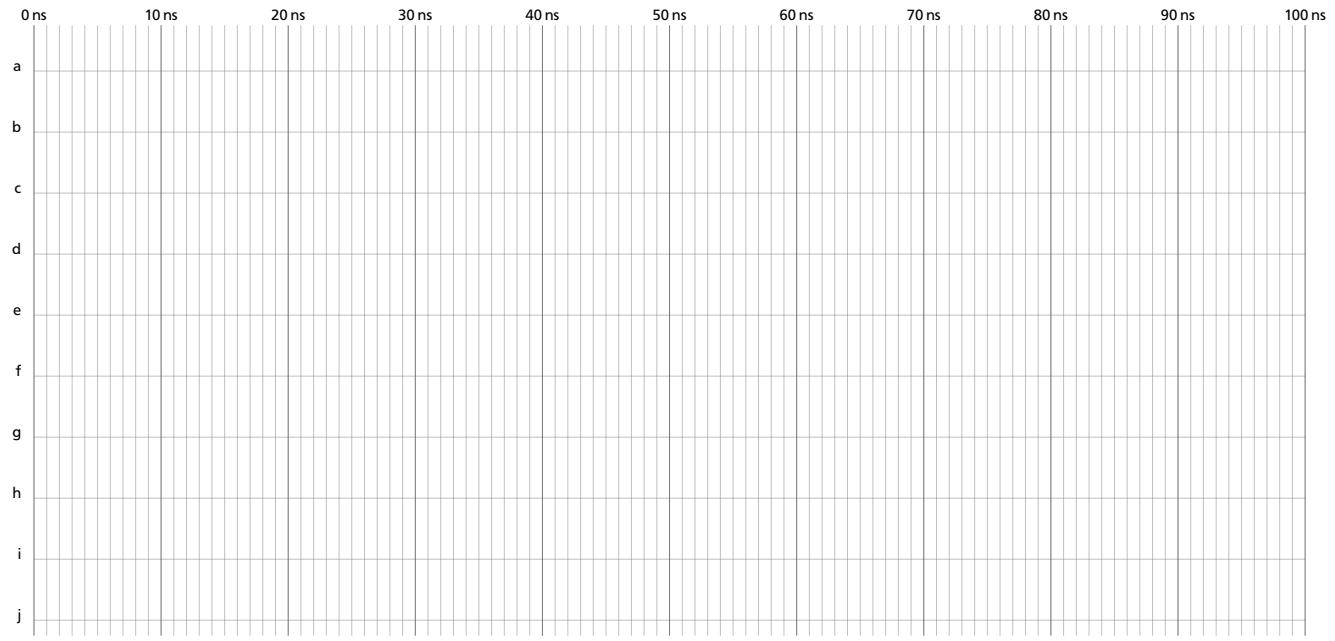
d) Welcher Vor- und welcher Nachteil ergibt sich hauptsächlich aus der Verwendung von vielen Pipeline-Stufen?

Übung 11.2 Zeitverhalten sequentieller Beschreibungen [15 min]

Simulieren Sie das Verhalten der nachfolgenden Signale für die ersten 100 ns. Bedenken Sie, dass bei einfachen **always** Blöcken (im Gegensatz zu **always_comb**) die Signalinitialisierung nicht als Signaländerung interpretiert wird.

seq/timing.sv

```
1  `timescale 1 ns / 10 ps
2  module timing;
3    localparam x = 2;
4
5    logic [2:0] a = 0;
6    always begin if (!a[0]) #10; else #(3+x); a <= a+1; end
7
8    logic b, c, d, e, f, g, h, i, j;
9    assign b = ^a;
10   always              begin             c   = b;     d   = c; @(negedge a[0]); end
11   always              begin             e   = b; #a; f   = e; @(posedge a[0]); end
12   always @(negedge b) begin     g <= c; h <= g;                               end
13   always @(f|d)       begin #2; i   = e; j <= i;                              end
14 endmodule
```



Übung 11.3 Parametrisierte Moduldefinition [15 min]

In dieser Aufgabe soll ein XOR Gatter mit einer variablen Anzahl an Eingängen realisiert werden.

a) Implementieren Sie ein funktionales Modul (Verhaltensbeschreibung) zu folgender Schnittstelle:

comb/xor/Parameter_Xor_Funct.sv

```
1 module xor_functional #(parameter SIZE = 2)
2                         (input  logic [SIZE-1 : 0] I,
3                          output logic              O);
```

b) Realisieren Sie ein äquivalentes strukturelles Modul (Strukturbeschreibung) zu nachfolgender Schnittstelle. Nutzen Sie dafür eine **for** Schleife um eine variable Anzahl an Zuweisungen zu generieren.

comb/xor/Parameter_Xor_Struct.sv

```
1 module xor_structural #(parameter SIZE = 2)
2                         (input  logic [SIZE-1 : 0] I,
3                          output logic              O);
```

c) Entwickeln Sie eine selbsttestende Testbench für beide Module mit Größe 4.