

25.05.2006

Technische Grundlagen der Informatik II

4. Übung – Mikroprogramme

Sommersemester 2006

Aufgabe 1: Division

Dividieren Sie die folgenden Dualzahlen nach der Vergleichsmethode und der Methode ohne Rückstellung des Restes (Vorlesung Kapitel 3, Folien 48 ff)

a) $0101010 \div 0101$

Lösung:

Vergleichsmethode:

$$\begin{array}{r} 0101010 \div 0101 = \quad 1000, \text{ Rest: } 010 \\ - \underline{0101} \quad \text{Passt} \mapsto 1 \\ \quad 00000 \\ (- \underline{0101}) \quad \text{Passt nicht} \mapsto 0 \\ \quad \quad 00001 \\ (- \underline{0101}) \quad \text{Passt nicht} \mapsto 0 \\ \quad \quad \quad 00010 \\ (- \underline{0101}) \quad \text{Passt nicht} \mapsto 0 \\ \quad \quad \quad \quad 0010 \quad \mapsto \quad \text{Rest: } 010 \end{array}$$

Methode ohne Rückstellung des Restes:

$$D = 0101, D' = 1011$$

$$\begin{array}{r} 0101010 \div 0101 = \text{Overflow} \\ + \underline{1011} \quad \mapsto +D', \text{ Test auf Overflow im ersten Schritt} \\ (1)0000 \quad \mapsto \text{nicht negativ, Overflow} \end{array}$$

b) $0101010 \div 0110$

Lösung:

Vergleichsmethode:

$$\begin{array}{r} 0101010 \div 0110 = 0111, \text{ Rest: } 000 \\ (-) \underline{0110} \quad \text{Passt nicht} \mapsto 0 \\ 01010 \\ - \underline{0110} \quad \text{Passt} \mapsto 1 \\ 01001 \\ - \underline{0110} \quad \text{Passt} \mapsto 1 \\ 00110 \\ - \underline{0110} \quad \text{Passt} \mapsto 1 \\ 0000 \quad \mapsto \text{ Rest: } 000 \end{array}$$

Methode ohne Rückstellung des Restes:

$$D = 0110, D' = 1010$$

$$\begin{array}{r} 0101010 \div 0110 = 0111, \text{ Rest: } 0000 \\ + \underline{1010} \quad \mapsto +D', \text{ Test auf Overflow im ersten Schritt} \\ 11110 \quad \mapsto \text{negativ, Quotientenbit ist 0, daher kein Overflow} \\ + \underline{0110} \quad +D \\ (1) 01001 \quad \mapsto \text{positiv, Quotientenbit ist 1} \\ + \underline{1010} \quad +D' \\ (1) 00110 \quad \mapsto \text{positiv, Quotientenbit ist 1} \\ + \underline{1010} \quad +D' \\ (1) 0000 \quad \mapsto \text{positiv, Quotientenbit ist 1} \\ 0000 \quad \mapsto \text{Rest} \end{array}$$

Aufgabe 2: Schrittsteuerwerk in Verilog

Das in der Vorlesung (Kapitel 4, Folie 18) angegebene Verilog-Programm ist zu vervollständigen:

```
module stw(clk, x1, x2, x3, y1, y2);
  input  clk, x1, x2, x3;
  output y1, y2;

  reg [3:0] z; // Zustandsregister sync

  initial z = 4'b0001;

  assign y1 = z[0]&x1 | z[1] | z[2]&x2;

  a) assign y2 = z[1] | z[2]&~x3 | z[3];

  always@(posedge clk)
  begin
    z[0] <= z[0]&~x1 | z[3]&(~x1|~x3);
    z[1] <= z[0]&x1;
```

```
b) z[2] <= z[1]&x2 | z[3]&x1 & x3;
```

c) `z[3] <= z[2] | z[1]&~x2;`

```
end
endmodule
```

Aufgabe 3: Mikroprogramm-Steuerwerk

Gegeben ist ein MP-Steuerwerk, ähnlich wie das aus der Vorlesung Kapitel 4, Folie 15.

```
module stw(clk,x1,y1);
  input  x1, clk;
  output y1;

  reg [1:0] s; initial s=0; // Zustandsregister sync
  reg [2:0] FG [0:7]; // ROM

  initial begin
    FG[0]=3'b001; FG[1]=3'b110; FG[2]=3'b110; FG[3]=3'b001;
    FG[4]=3'b011; FG[5]=3'b100; FG[6]=3'b111; FG[7]=3'b010;
  end

  wire [2:0] bef;

  assign bef = FG[{x1,s}];
  assign y1 = bef[0];

  always @(posedge clk) s <= bef[2:1];
endmodule
```

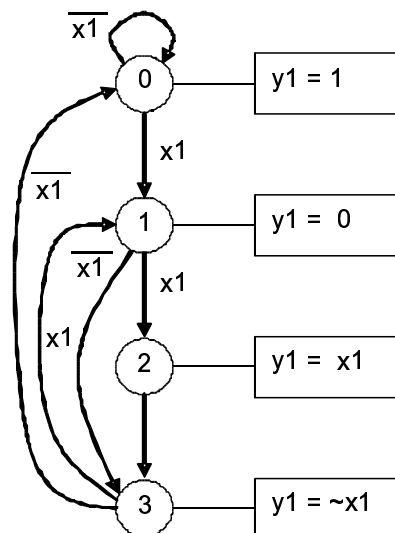
a) Geben Sie die Übergangstabelle und den Zustandsgraphen an. Sie können die Zustände dabei zur Vereinfachung auch dezimal darstellen.

Lösung:

Übergangstabelle:

x1	s	s'	y1
0	0	0	1
0	1	3	0
0	2	3	0
0	3	0	1
1	0	1	1
1	1	2	0
1	2	3	1
1	3	1	0

Zustandsgraph:



b) Implementieren Sie das gleiche Steuerwerk als Zustandsautomat in Verilog

Lösung:

```
module stw_statemachine(clk, reset, x1, y1);
  input  clk, reset, x1;
  output y1;

  reg [1:0] z; // Zustandsregister sync

  assign y1 = (z==0) || ((z[1]==1) && (z[0]^x1));

  always @(posedge clk or posedge reset) begin
    if (reset) begin // asynchroner Reset
      z<=2'b00;
    end
    else begin
      case(z)
        0: if (x1) z<=1; else z<=0;
        1: if (x1) z<=2; else z<=3;
        2: z<=3;
        3: if (x1) z<=1; else z<=0;
      endcase
    end
  end
end
endmodule
```