Einführung in Computer Microsystems

TECHNISCHE UNIVERSITÄT DARMSTADT

- Wiederholung / Klausurvorbereitung

Klausurinfos

TECHNISCHE UNIVERSITÄT DARMSTADT

- Hier schonmal vorab die Spielregeln bei der Klausur
- Es wird in den Teilklausuren je 60 Punkte geben
- Die Punkte werden einfach addiert (120 Punkte)
- Zum Bestehen werden 60 Punkte (50%) benötigt

1. Teilklausur

TECHNISCHE UNIVERSITÄT DARMSTADT

- 1.6.12, 18:00 - 19:00
- Bearbeitungszeit 60min
- Zum leichteren Verständnis/Einfinden in die Aufgabenstellung werden wir die Klausur zu Beginn vorlesen
- Während dieser Zeit sind keine Fragen erlaubt
- Während dieser Zeit ist keine Kommunikation erlaubt
- Während dieser Zeit ist kein Schreiben erlaubt

Welcher Stoff ist für die Klausur relevant

TECHNISCHE UNIVERSITÄT DARMSTADT

- Alle bisherigen Vorlesungen (natürlich nicht der ARM Vortrag)
- Uneingeschränkt die Übungsblätter 1 und 3
- Übungsblatt 2 mit Ausnahme der bisher in der Vorlesung nicht behandelten Themen (FSM, Busse)

Hilfsmittel

TECHNISCHE UNIVERSITÄT DARMSTADT

- Sie benötigen: Stift / Lineal / Getränke / Ausweis
- Ein Verilog-Syntaxblatt wird an die Klausur angehängt sein
- Ansonsten sind KEINE Hilfsmittel erlaubt

Raumeinteilung nach Nachnamen

TECHNISCHE UNIVERSITÄT DARMSTADT

- A - E: S101/A01 (70 Studenten)
- F - J: S202/C205 (63 Studenten)
- K - L: S101/A03 (45 Studenten)
- M - Z: S101/A1 (169 Studenten)

- Bitte S101/A01 und S101/A1 nicht verwechseln
- M - Z sind im großen Raum

Ergebnisse

TECHNISCHE UNIVERSITÄT DARMSTADT

- Punkte werden im Moodle eingetragen
- Korrektur wird etwas Zeit benötigen
- Noten erst nach der zweiten Teilklausur

Hausaufgaben - Mittelwert

TECHNISCHE UNIVERSITÄT DARMSTADT

```
module mittelwert(
  input  wire [15:0] in1,
  input  wire [15:0] in2,
  input  wire [15:0] in3,
  input  wire [15:0] in4,
  output wire [15:0] out
  );

  //18 Bit wegen möglichem Überlauf
  wire [17:0] temp;

  // Alle vier Werte zusammen addieren.
  assign temp = in1 + in2 + in3 + in4;

  // /4 durch shiften oder direkte Auswahl der Bits
  assign res = temp[17:2];

endmodule
```

Hausaufgaben - Mittelwert

TECHNISCHE UNIVERSITÄT DARMSTADT

```
module tb()
  reg  [15:0] in1,in2,in3,in4;   //Eingänge als reg
  wire [15:0] out;               //Ausgänge als wire

  //Instanz
  mittelwert  uut (
    .in1(in1),
    .in2(in2),
    .in3(in3),
    .in4(in4),
    .out(out)
  );

  //Stimulus
  initial begin
    // alle Werte 0 --> erwartet 0
    in1 = 0;  in2 = 0;  in3 = 0;  in4 = 0; #10
    // normale Eingaben --> erwartet 10
    in1 = 10;  in2 = 5;  in3 = 5;  in4 = 20;  #10
    ..............
  end
endmodule
```

Hausaufgaben - CRC

TECHNISCHE UNIVERSITÄT DARMSTADT

```
module crc(
  input  wire          clk,      //Takt
  input  wire          reset,    //Reset
  input  wire          datain,   //Eingang der Daten, werden seriell angelegt
  input  wire          enable,   //= 1, wenn gültige Daten am Eingang liegen
  input  wire          check,    //0 = generiere CRC, 1 = checke CRC
  output wire [4:0]    dataout,  //Ausgang der Daten im Generierungsmodus
  output wire          error     //= 1, falls ein Fehler im Checkmodus festgestellt wird
  )
```

Implementierung mittels Shift-Register

Pseudocode - CRC

TECHNISCHE UNIVERSITÄT DARMSTADT

```
Schieberegister := 0000 (Startwert)
 solange Bits im String verbleiben:
   falls  das am weitesten links stehende Bit vom Schieberegister
          ungleich dem nächsten Bit aus dem String ist:
     Schieberegister := (Schieberegister linksschieben um 1, rechtes Bit 0)
                        xor CRC–Polynom
   andernfalls:
     Schieberegister := Schieberegister linksschieben um 1, rechtes Bit 0
   nächstes Bit im String
 Schieberegister enthält das Ergebnis.
```

Hausaufgaben - CRC

TECHNISCHE UNIVERSITÄT DARMSTADT

```
    //Shiftregister
    reg [4:0] shiftreg;

    //CRC-Polynom, kann auch unten direkt drinnen stehen
    wire [5:0] crc_poly = 6'b101001;

    always @(posedge clk) begin
      //Reset --> Shiftregister zuruecksetzen
      if(reset) begin
        shiftreg <= 6'b000000;
      end
      else begin
        //Wenn enable --> CRC berechnen
        if(enable) begin
          //wenn shiftreg[4] != dem Eingang, dann shiften und XOR
          if(shiftreg[4] != datain)
            shiftreg[4:0] <= {shiftreg[3:0], 1'b0} ^ crc_poly[4:0];
          //ansonsten nur shiften
          else
            shiftreg[4:0] <= {shiftreg[3:0], 1'b0};
        end
      end
    end

    //Ausgang zuweisen
    assign dataout = shiftreg;

    //Fehler, falls Check-Modus und Shiftregister ungleich 0
    assign error = (check & (|shiftreg));
endmodule
```

Hausaufgabe - kasakdierter Multiplizier

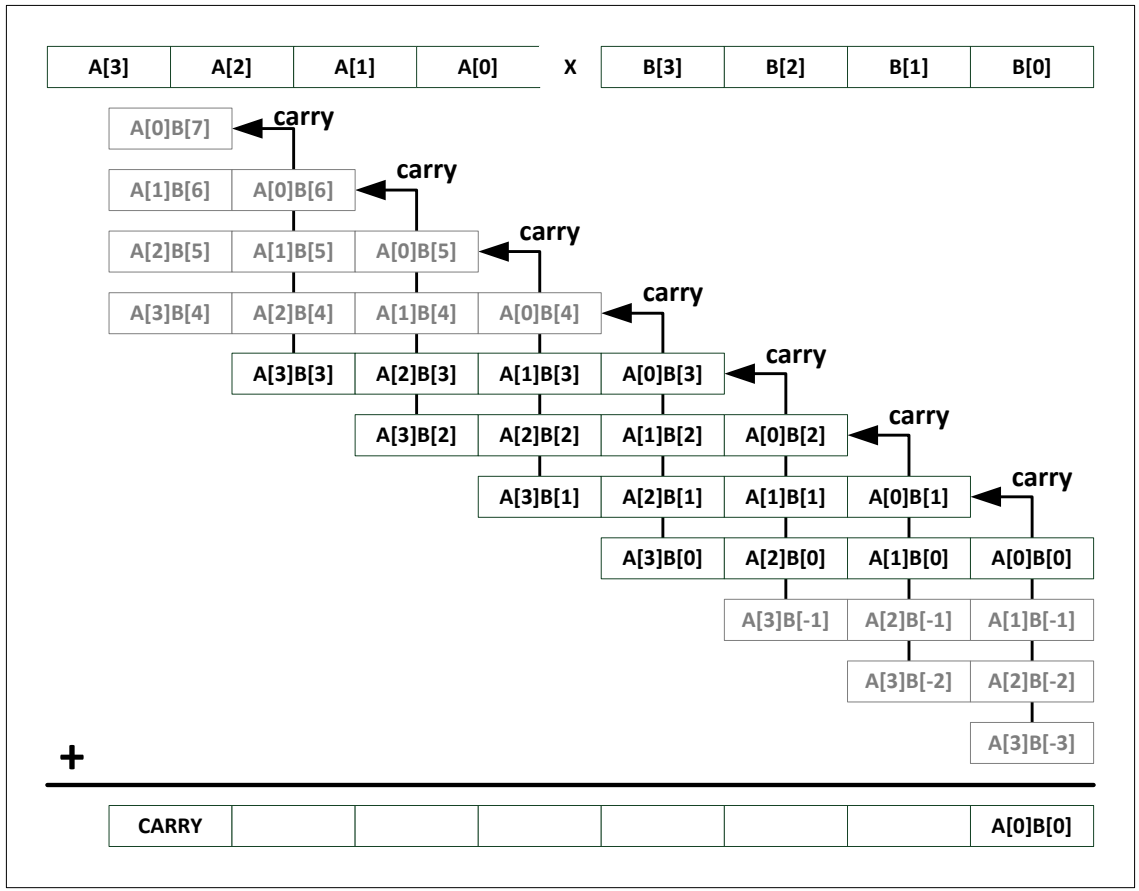
TECHNISCHE UNIVERSITÄT DARMSTADT

```
module multiplier (
  input  wire [3:0] A,      // Operand A
  input  wire [3:0] B,      // Operand B
  output wire [7:0] R);     // Ergebnis der Mulitplikation R


module full_adder(
  input  wire  A,       // Operand A
  input  wire  B,       // Operand B
  input  wire  C_IN,    // eingehendes Carry
  output wire  C_OUT,   // ausgehendes Carry
  output wire  SUM);    // Summe aus A+B
```

Hier jetzt eine generate Variante

TECHNISCHE UNIVERSITÄT DARMSTADT



```
wire [40:0] c;                      // Wire Definition für die Carries
wire [39:0] s;                      // Wire Definition für die Zwischensummen
wire [ 3:0] a = A;                  // Zuweisung von Operand A
wire [11:0] b = {4'h0, B, 4'h0};    // Zuweisung von Operand B

assign R = s[39:32];                // Zuweisung der Endsummen auf
assign c[0]   = 1'b0;               // kein eingehendes Carry
assign s[7:0] = 8'h00;              // keine eingehende Zwischensummen

genvar i, k;

generate
  for(i=0;i<4;i=i+1) begin
    for(k=0;k<8;k=k+1) begin
      full_adder add_ik (.A     (a[i] & b[4+k-i]),
                         .B     (s[8*i+k]),
                         .C_IN  (c[8*i+k]),
                         .C_OUT (c[8*i+k+1]),
                         .SUM   (s[8*(i+1)+k]));
    end
  end
endgenerate
```

Fragen

TECHNISCHE UNIVERSITÄT DARMSTADT

- for vs. generate
- Synthetisierbarkeit von ’/’ Operator
- Werte in mehreren always-Blöcken zuweisen

noch Fragen?

TECHNISCHE UNIVERSITÄT DARMSTADT

- jetzt stellen
- Sprechstunden heute / morgen und auch Freitag
- ins Forum stellen

Viel Erfolg bei der Klausur!