

# Architekturen und Entwurf von Rechnersystemen



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Beispielaufgaben

---

## Architekturen und Entwurf von Rechnersystemen Beispielaufgaben

Name, Vorname: \_\_\_\_\_ Matrikelnummer: □□□□□□□□

---

### Hinweise zur Klausur:

- Bitte lesen Sie die Hinweise zu Beginn der Klausur!
- Füllen Sie JETZT das Deckblatt VOLLSTÄNDIG aus!
- Schalten Sie Ihr Handy aus.
- Legen Sie Studentenausweis und ein Lichtbildausweis zur Kontrolle bereit.
- Verwenden Sie nur blaue oder schwarze dokumentenechte Stifte, kein Rot oder Grün, keine Bleistifte.
- Die Heftung der Klausur darf nicht gelöst werden. Ausnahme ist das Bluespec-Syntaxblatt auf der letzten Seite.
- Packen Sie alles bis auf Schreibwerkzeug weg. Es sind KEINE Hilfsmittel erlaubt (Ausnahme: Ein Wörterbuch für ausländische Studierende). Wird während der Klausur ein unerlaubtes Hilfsmittel gefunden, wird dies als Täuschungsversuch gewertet und die Prüfung gilt als nicht bestanden. In schweren Fällen von Täuschung behalten wir uns weitere Schritte bis hin zur Exmatrikulation vor.
- Essen und Trinken sind erlaubt, nehmen Sie jedoch Rücksicht auf Ihre Kommilitonen.
- Bewertet wird der Lösungsweg, nicht nur das Ergebnis. Schreiben Sie alle Zwischenschritte auf, damit die Lösung nachvollziehbar ist.
- Falls Sie Fragen zur Aufgabenstellung haben, melden Sie sich bitte. Das Aufsichtspersonal wird dann an Ihren Platz kommen. Inhaltliche Fragen werden nicht beantwortet.
- Falls Sie zusätzliches Papier benötigen, melden Sie sich bitte. Sie erhalten dann Papier von uns, eigenes Papier ist nicht gestattet. Beschriften Sie die Zusatzblätter mit Namen und Matrikelnummer.
- Falls Sie auf Toilette müssen, kommen Sie bitte mit Ihrer Klausur nach vorne und geben sie beim Aufsichtspersonal ab.

Viel Erfolg!

---

## Architekturen und Entwurf von Rechnersystemen Beispielaufgaben

Name, Vorname: \_\_\_\_\_ Matrikelnummer: □□□□□□□□

---

### Aufgabe 1 Bluespec Allgemein

(10 Punkte)(2+2+2+2+2)

---

- a) Geben Sie ein kleines ( $\leq 5$  Zeilen) Beispiel für die Benutzung des Bluespec Zuweisungsoperators  $\leftarrow$ .
- b) Geben Sie ein kleines ( $\leq 5$  Zeilen) Beispiel für die Benutzung des Bluespec Zuweisungsoperators  $\leftarrow$ .
- c) Beschreiben Sie in maximal zwei Sätzen, wann der Bluespec Zuweisungsoperator  $=$  verwendet wird.
- d) Nennen Sie zwei fundamentale Eigenschaften einer Bluespec Regel.
- e) Beschreiben Sie in maximal zwei Sätzen, wie sich `method ActionValue#(UInt#(32))` von `method UInt#(32)` unterscheidet.

## Architekturen und Entwurf von Rechnersystemen Beispielaufgaben

Name, Vorname: \_\_\_\_\_ Matrikelnummer: □□□□□□□□

### Aufgabe 2 Space-Latency Tradeoff

(9 Punkte)(5 + 4)

Gegeben sei das folgende Bluespec Modul:

```
1  typedef Server#(Tuple2#(Int#(32), Int#(32)), Int#(32)) PythagorasServer;
2  module mkCalcPythagoras(PythagorasServer);
3      FIFO#(Tuple2#(Int#(32), Int#(32))) in <- mkFIFO();
4      FIFO#(Int#(32)) out <- mkFIFO();
5      Server#(Int#(32), Int#(32)) sqrt <- mkSqrt(); // Module to calculate Square Root
6      rule calc; // c = sqrt(a^2 + b^2)
7          let val = in.first(); in.deq();
8          let a = tpl_1(val);
9          let b = tpl_2(val);
10         let aSqr = a * a;
11         let bSqr = b * b;
12         sqrt.request.put(aSqr + bSqr);
13     endrule
14     rule fetchResult;
15         let val <- sqrt.response.get();
16         out.enq(val);
17     endrule
18     interface Put = toPut(in);
19     interface Get = toGet(out);
20 endmodule
```

- a) Instantiieren Sie das oben definierte Modul. Setzen Sie einen Request ab. Definieren Sie sich dafür einen passenden Eingabewert mit let. Holen Sie sich eine Response vom Modul und geben Sie das Ergebnis mit \$display aus.

```
1  // Instantiierung:
2
3
4
5
6
7
8  // Request:
9  rule put;
10
11
12
13
14
15
16
17  endrule
```

---

## Architekturen und Entwurf von Rechnersystemen Beispielaufgaben

Name, Vorname: \_\_\_\_\_ Matrikelnummer: □□□□□□□□

---

```
1 // Response:
2 rule get;
3
4
5
6
7
8
9
10
11 endrule
```

- b) Welches Problem ergibt sich bei diesem Modul in Hinblick auf den kritischen Pfad? Begründen Sie in maximal zwei Sätzen.

---

## Architekturen und Entwurf von Rechnersystemen Beispielaufgaben

Name, Vorname: \_\_\_\_\_ Matrikelnummer: □□□□□□□□

---

### Aufgabe 3 Bluespec Regeln

(20 Punkte)(2 + 4 + 4 + 10)

Diese Aufgabe beschäftigt sich mit der Regelauswertung in Bluespec.

a)

```
1  module mkConflict(Empty);
2      Reg#(Int#(32)) x <- mkReg(0);
3      Reg#(Int#(32)) y <- mkReg(0);
4
5      rule writeX;
6          x <= y + 5;
7      endrule
8
9      rule writeY;
10         y <= x * x;
11     endrule
12 endmodule
```

In diesem Modul gibt es Konflikte. Begründen Sie diese mit den Scheduling-Eigenschaften von Bluespec Regeln in maximal zwei Sätzen.

b) Lösen Sie den Konflikt aus Teilaufgabe a) auf. In jedem Takt soll  $x = y + 5$  und  $y = x * x$  ohne vorgegebene Reihenfolge berechnet werden.

```
1  module mkSolveConflict(Empty);
2      Reg#(Int#(32)) x <- mkReg(0);
3      Reg#(Int#(32)) y <- mkReg(0);
4
5
6
7
8
9
10
11
12
```

## Architekturen und Entwurf von Rechnersystemen Beispielaufgaben

Name, Vorname: \_\_\_\_\_ Matrikelnummer: □□□□□□□□

```
13
14
15
16
17
18
19
20  endmodule
```

c) Die Berechnung von  $x = y + 5$  soll mit Hilfe eines Taktzählers nur noch alle zwei Takte ausgeführt werden.

```
1  module mkEverySecond(Empty);
2      Reg#(Int#(32)) x <- mkReg(0);
3      Reg#(Int#(32)) y <- mkReg(0);
4
5      Reg#(UInt#(1)) counter <- mkReg(0);
6      rule cnt;
7          counter <= counter + 1;
8      endrule
9
10     rule writeX (counter == 0);
11         x <= y + 5;
12     endrule
13
14     rule writeY;
15         y <= x * x;
16     endrule
17 endmodule
```

Bei der Simulation des Moduls ergibt sich folgende Waveform:

Signalname	Signalverlauf
CLK	
CAN_FIRE_RL_writeX	
WILL_FIRE_RL_writeX	
CAN_FIRE_RL_writeY	
WILL_FIRE_RL_writeY	
counter	
x[31:0]	00000000
y[31:0]	00000000

---

## Architekturen und Entwurf von Rechnersystemen Beispielaufgaben

Name, Vorname: \_\_\_\_\_ Matrikelnummer: □□□□□□□□

---

Warum ändert sich der Stand von  $x$  und  $y$  nicht?

Welches Scheduling-Attribut können Sie verwenden damit das Modul das korrekte Verhalten zeigt?

d) Implementieren Sie einen Zähler mit folgenden Eigenschaften:

- Der Zählerstand kann durch die Methoden `up` (Zählerstand um  $x$  erhöhen), `down` (Zählerstand um  $x$  senken) und `reset` (Zählerstand = 0) manipuliert werden.
- Alle Methoden können im selben Takt auftreten und müssen korrekt behandelt werden.
- Die Ausgabemethode soll den momentanen Zählerstand unter Berücksichtigung aller im selben Takt erfahrenen Veränderungen liefern.

Hinweis: Sie können das `CReg` Modul aus der Vorlesung verwenden.

```
1 interface Counter;
2     method Action up(Int#(32) v);
3     method Action down(Int#(32) v);
4     method Action reset();
5     method Int#(32) getCounter();
6 endinterface
7
8 module mkCounter(Counter);
9
10
11
```



---

## Architekturen und Entwurf von Rechnersystemen Beispielaufgaben

Name, Vorname: \_\_\_\_\_ Matrikelnummer:

---

12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64

```
method Action reset();
```

```
endmethod
```

```
method Action up(Int#(32) v);
```

```
endmethod
```

```
method Action down(Int#(32) v);
```

---

## Architekturen und Entwurf von Rechnersystemen Beispielaufgaben

Name, Vorname: \_\_\_\_\_ Matrikelnummer: □□□□□□□□

---

```
65
66
67
68
69     endmethod
70
71
72
73     method Int#(32) getCounter();
74
75
76
77
78
79
80
81
82
83     endmethod
84
85
86
87
88
89
90
91
92
93
94     endmodule
```

## Architekturen und Entwurf von Rechnersystemen Beispielaufgaben

Name, Vorname: \_\_\_\_\_ Matrikelnummer: □□□□□□□□

### Aufgabe 4 Erweiterte Bluespecaufgaben

(14 Punkte)(8 + 6)

a) Geben Sie die korrekten Provisos für folgendes Modul an. Der Eingabetyp `maximumValue` gibt den höchsten zu verarbeitenden Wert an.

Hinweise:

1. Erzeugen Sie mit Hilfe der Provisos die Typen `operandsBits` und `resultBits`.
2. Die Bitbreite des Ergebnisses einer Multiplikation ist höchstens die Summe der Bitbreite der Operanden.
3. Stellen Sie sicher, dass `resultBits` kleiner oder gleich 64 bit ist.
4. Stellen Sie sicher, dass `operandsBits` kleiner oder gleich `resultBits` ist.

```
1  interface Mult#(numeric type maximumValue);
2      method Action placeOperands(Int#(TLog#(maximumValue)) a,
3                                  Int#(TLog#(maximumValue)) b);
4      method Int#(64) getResult();
5  endinterface
6
7  module mkMult(Mult#(maximumValue))
8      provisos(
9
10
11
12
13
14
15
16
17
18          );
19
20      Reg#(Int#(resultBits)) result <- mkReg(0);
21
22      method Action placeOperands(Int#(operandsBits) a, Int#(operandsBits) b);
23          Int#(resultBits) aE = extend(a);
24          Int#(resultBits) bE = extend(b);
25
26          result <= aE * bE;
27      endmethod
28
29      method Int#(64) getResult();
30          return extend(result);
31      endmethod
32
33  endmodule
```

---

## Architekturen und Entwurf von Rechnersystemen Beispielaufgaben

Name, Vorname: \_\_\_\_\_ Matrikelnummer: □□□□□□□□

---

- b) Erklären Sie kurz, wofür Typklassen eingesetzt werden und geben Sie drei Beispiele von Typklassen in Bluespec an.

---

## Architekturen und Entwurf von Rechnersystemen Beispielaufgaben

Name, Vorname: \_\_\_\_\_ Matrikelnummer: □□□□□□□□

---

### Aufgabe 5 System-on-Chip

(12 Punkte)(4 + 4 + 2 + 2)

a) Was versteht man unter einem System-on-Chip? Welche Komponenten findet man gewöhnlich in einem solchen?  
(Maximal 4 Sätze)

b) Welche Speicher sind auf dem Zynq 7000 SoC vorhanden?

c) Welche Art von Rechenarchitektur setzt die NEON-Einheit von ARM-Prozessoren um?

---

## Architekturen und Entwurf von Rechnersystemen Beispielaufgaben

Name, Vorname: \_\_\_\_\_ Matrikelnummer: □□□□□□□□

---

- d) Nennen Sie zwei Nachteile von Softcore-Prozessoren gegenüber Hardcore-Prozessoren in FPGA Systemen.

---

## Architekturen und Entwurf von Rechnersystemen Beispielaufgaben

Name, Vorname: \_\_\_\_\_ Matrikelnummer: □□□□□□□□

---

### Aufgabe 6 Advanced eXtensible Interface Bus (AXI)

(18 Punkte)(2 + 4 + 3 + 2 + 7)

---

a) Welche Funktion hat AXI in einem System on Chip?

b) Was ist der wesentliche Unterschied zwischen AXI ACP und AXI HP Ports auf Zynq SoCs?

c) Beschreiben Sie stichpunktartig folgende AXI Arten:

- AXI4:

- AXI4-Lite:

## Architekturen und Entwurf von Rechnersystemen Beispielaufgaben

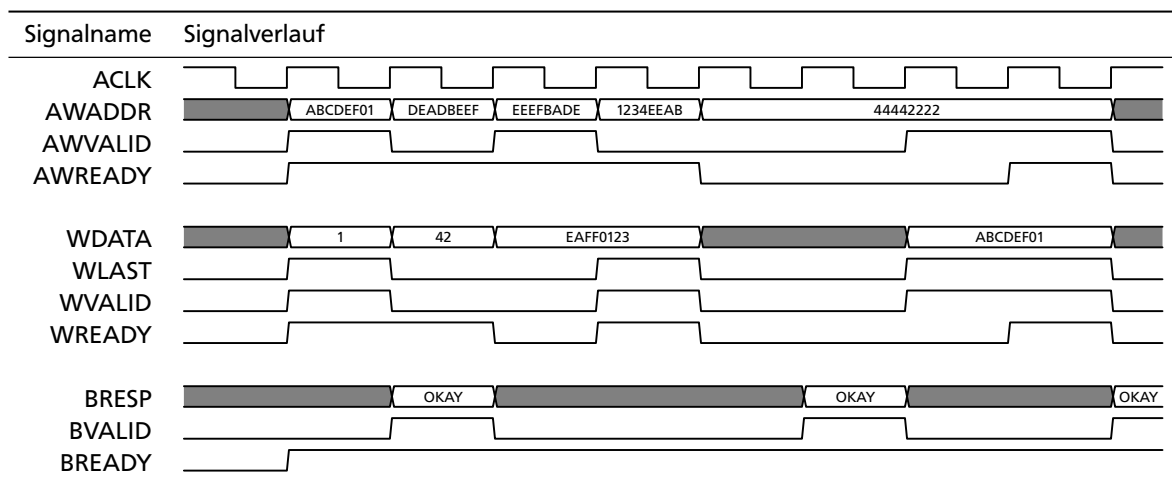
Name, Vorname: \_\_\_\_\_ Matrikelnummer: □□□□□□□□

- AXI4-Stream:

d) Warum bietet AXI die Möglichkeit Burst-Transfers zu veranlassen? (Maximal zwei Sätze)

e) Welche Daten werden bei diesem (vereinfacht dargestellten) AXI-Transfer an welche Speicherstelle übertragen? Tragen Sie die Kommunikationsrichtung sowie die Transfers in die Tabelle ein.

- AWLEN: 0
- Wortbreite: 32 bit
- Burst Modus: Incremental





---

## Architekturen und Entwurf von Rechnersystemen Beispielaufgaben

Name, Vorname: \_\_\_\_\_ Matrikelnummer: □□□□□□□□

---

Kommunikationsrichtung:

Transfer Adresse

Daten

---