

## Hardware-Design Praktikum

# Fragenkatalog

WS 2017/18

1. Wie funktioniert ein FPGA grundsätzlich?
2. Nehmen Sie Ihre Matrikelnummer und rechnen Sie diese Dezimalzahl in das Binär-, Oktal- und Hexadezimalsystem um. Notieren Sie Ihre Matrikelnummer im Hexadezimalsystem als „Little Endian“ und als „Big Endian“.
3. Bestimmen Sie jeweils den Zahlenbereich für  $u_{10}$ ,  $b_{v11}$ ,  $b_{1,12}$  und  $b_{2,13}$ . Nennen Sie außerdem jeweils einige Vor- bzw. Nachteile dieser Zahlendarstellungen.
4. Wieviele Speicherstellen kann ein Adressbus mit 16 Leitungen direkt adressieren? Begründung?
5. Was ist komplementär zu  $\overline{(a \wedge b)} \vee \overline{(a \vee b)}$ ?
6. Welcher Ausdruck ist ohne Vereinfachungen dual zu  $\bar{a} \vee b$ , welcher zu  $1 \wedge 0$ ?
7. Wieso verwendet man unterschiedliche Speichertypen wie Cache, RAM und Register in einem Rechner?
8. Was unterscheidet in VHDL eine Variable und ein Signal?
9. Was bestimmt die Sensitivitätsliste eines process-Statements in VHDL? Gibt es in VHDL auch process-Statements ohne Sensitivitätsliste? Betrachten Sie außerdem folgendes VHDL-Programmfragment

```
entity recapitulationCircuit is
end recapitulationCircuit;

architecture behavior of recapitulationCircuit is
  signal a,b,c,d,e : bit;
  begin
    process (a,b)
    begin
      c <= a or b after 5 ns;
      d <= b and c after 6 ns;
    end process;

    process (a,c)
    begin
      e <= transport a xor d after 3 ns;
    end process;
  end architecture behavior;
```

Ergebnis sofort berechnen, mit 3ns verzögerung ausgeben

und geben Sie zu den folgenden Signalverläufen für  $a$  und  $b$  die resultierenden Signale  $c$ ,  $d$  und  $e$  bis zu einer Simulationsdauer von 90 ns an, wobei  $c$ ,  $d$  und  $e$  den Anfangswert 0 besitzen:

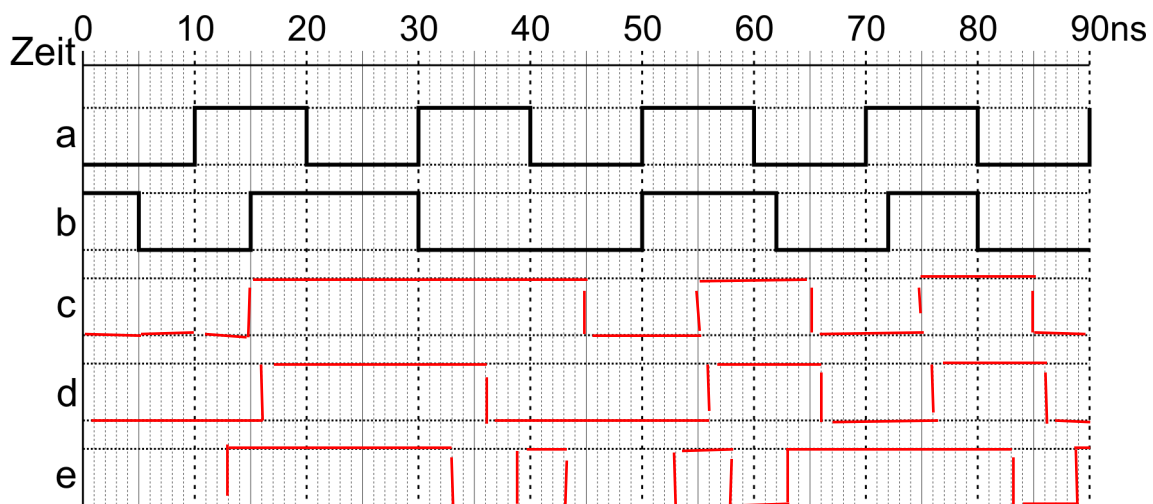


Abbildung 1: (unvollständiges) Timing-Diagramm zum gegebenen Programmfragment

10. Wie groß ist der Programmspeicher PMEMORY in kB (Kilobyte)?
11. Wieso reicht hier eine Adressbreite von 12 Bit?

12. Wieso verwendet man nicht möglichst viele Register?
13. Welche Vorteile, welche Probleme bringt die Verwendung des Nullregisters mit sich?
14. Zu welchem Zweck werden die Befehle GETOV und SETOV benötigt? **Tip:** Betrachten Sie hierzu auch den Befehl BOV in der Befehlsübersicht in Anhang ??!
15. Vergleichen Sie die Datenpfad-Schaltung mit dem Zustandsübergangsdiagramm der Kontrolle (Abb. ??). Welche Zwischenregister werden in den einzelnen Zuständen beschrieben?
16. Wie funktioniert die Vorzeichenexpansion bei  $u_n$ ,  $bv_n$ ,  $b_{1,n}$  und  $b_{2,n}$  korrekt? Und warum wird der Immediate-Operand bei einigen Befehlen überhaupt vorzeichenexpandiert?
17. Wozu gibt es den PassImmed-Opcode?
18. Was ist der Unterschied zwischen einem *Moore*- und einem *Mealy*-Automaten? Welchen Automaten zeigt Abbildung ???
19. Wieviele Takte werden für die Abarbeitung eines beliebigen Befehls (ohne Interrupts!) maximal benötigt? Wieviele Takte benötigt nun die Abarbeitung eines rein Register-basierten ADD-Befehls mindestens, wieviele Takte maximal?
20. Wie ändert sich die Abarbeitung eines Befehls, falls ein Interrupt auftritt?
21. Welche Stufe im Übergangsdiagramm aus Abbildung ?? bildet Ihrer Meinung nach den Flaschenhals bei der Befehlsverarbeitung? Begründung? Inwiefern können Multizyklus-Stufen bzw. Pipelining benutzt werden, um die Befehlsverarbeitung zu beschleunigen?
22. Bedeutet eine Verdoppelung der Taktrate bei dieser HaDesXI-Architektur auch eine Verdoppelung der Rechenleistung?
23. Was versteht man unter einem *kombinatorischen Schaltkreis*?
24. Warum kann beim JREG-Befehl der Opcode des CSHL-Befehls ausgegeben werden, obwohl doch der A-Operand unverändert durch die ALU an den PCBLOCK geleitet werden muss?
25. Warum wird der ADD-Opcode bei STORE- und LOAD-Befehl ausgegeben?  
**Tip:** Schauen Sie sich die Befehlssemantik in Anhang ?? an.
26. Wieso erscheinen für die ALU z.B. die beiden Befehle ENI und DPMA als NOP-Opcode?
27. Was geschieht mit dem Immediate-Operanden bei BOV, JAL und SISA?
28. Warum muss bei den Befehlen IN und LOAD die BOPADR nicht 000 sein?
29. Warum wird bei den Befehlen BNEZ und BEQZ nicht einfach der PassImmed-Opcode angelegt?
30. Weshalb ist es vorteilhaft, mehrere Interruptlevel zu unterstützen?
31. Warum ist es sinnvoll, für Ereignisse bei Peripheriegeräten Interrupts zu verwenden und nicht aktiv darauf zu warten, also aktiv den Zustand einer Komponente (periodisch) abzufragen (sog. *polling*)?
32. Wieviele Interruptereignisse speichert der HaDesXI-Prozessor bei deaktivierten Interrupts maximal?
33. Wie müsste man die IRQ-Logic ändern, wollte man 6 Interruptlevel unterstützen?
34. Wozu wäre ein *InvalidOpcode*-Interrupt nützlich?
35. Welcher Unterprogrammtaktik (*Caller Saves* bzw. *Callee Saves*) muss eine ISR folgen, und warum?
36. Wieviele XBus-Peripherie-Komponenten können maximal verwendet werden, wenn nur 8 Bit für deren Port-Adressen zur Verfügung stehen?