

Entwurf digitaler Systeme

Übungsblatt 3

8. Dezember 2017

1. Gegeben seien folgende VHDL-Fragmente:

```
signal a, b, c, d : integer;
...
process (a,b,c)
begin
    b <= a;
    c <= b;
    d <= b + c;
end process
```

```
signal a : integer;
...
process (a)
variable b, c, d : integer;
begin
    b := a;
    c := b;
    d := b + c;
end process;
```

- Wo liegt der Unterschied im Verhalten der beiden Listings?
 - Was passiert während der Simulation?
 - Was ändert sich, wenn im linken Listing die Signale b und c aus der Sensitivitätsliste entfernt werden?
 - Was ändert sich, wenn die Prozesse in getaktete Prozesse umgeschrieben werden?
 - Zeichnen Sie ein Blockdiagramm für die sich aus den getakteten Prozessen ergebenden Schaltungen.
2. Implementieren Sie einen 8-Bit Serial-to-Parallel Konverter.
3. In den kommenden Übungen soll ein Taschenrechner implementiert werden. Der Taschenrechner soll zunächst über vier Operationen (Addition, Subtraktion, Multiplikation, Division) verfügen. Darüber hinaus soll er die Negation des gegenwärtigen Operanden sowie Klammerung, Operationspriorisierung und das Speichern von 8 Ergebnissen unterstützen. Das Design soll in mehrere Blöcke unterteilt werden. Jeweils ein Block ist für die Eingabe und die Anzeige zuständig, ein weiterer für die Speicherung der Operanden, Ergebnisse und der aktuellen Eingabe. Schließlich übernimmt eine Einheiten die Berechnung der Ergebnisse und eine weitere die Steuerung.
- Die Eingabe erfolgt über 5 Pins (K[4:0]). Ein zusätzlicher Ausgabepin KE zeigt an, ob eine Eingabe getätigt werden darf.
 - Die Ausgabe erfolgt über 11 Siebensegmentanzeigen, die als Schieberegister angesprochen werden.

Codierung	Beschreibung
0x00	keine Eingabe
0x01 - 0x0a	Eingabe der Zahlen 0-9
0x0b	Negation des Operanden
0x0c - 0x0f	Operationen '+', '-', '*', '/'
0x10 - 0x11	Klammern '(', ')'
0x12	Ergebnis Anzeigen '='
0x13	Löschen der letzten Stelle
0x14	Löschen der Eingabe
0x15	Zurücksetzen
0x16	Speichern des Ergebnisses (gefolgt von einer Zahl von 1-8)
0x17	Laden des Ergebnisses (gefolgt von einer Zahl von 1-8)

Pin	Beschreibung
DR	Rücksetzen der Anzeige (Synchrones, high-aktives Reset)
DE	Übernahme von D und Rechtsshift
D[3:0]	Kodierung der Zahlen und des Negationszeichens

Die Ausgabe erfolgt Digitweise, d.h., es wird ein ganzes Zeichen über den 4-Bit-Datenbus übertragen. Die Codierung der Zeichen ist wie folgt definiert:

DR[3:0]	Beschreibung
0x0-0x9	Zahlen von 0-9
0xa	Vorzeichen '-'
0xb	Blank ' '
0xc	'E' zur Anzeige eines Fehlers

Die Ausgabe erfolgt immer als Sequenz von 11 Stellen, wobei das niederwertigste Digit (Zahlen von 0-9) zuerst ausgegeben wird. Belegt die Zahl nicht alle 11 Stellen, werden die restlichen Stellen mit Blanks ' ' aufgefüllt. Abbildung 1 zeigt die entsprechende Ausgabesequenz:

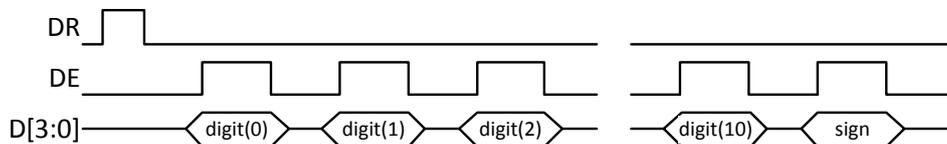


Abbildung 1: Ausgabesequenz

- Implementieren Sie das Design modular und überlegen Sie sich, welche Schnittstellen notwendig sind. Orientieren Sie sich an dem folgenden Blockschaltbild in Abbildung 2. Das Blockschaltbild ist nicht zwingend vollständig. Überlegen Sie sich, welche zusätzlichen Signale benötigt werden.
- Implementieren Sie das Design Schritt für Schritt, d.h., realisieren Sie das Design erst einmal ohne Klammerung und Operationspriorisierung.

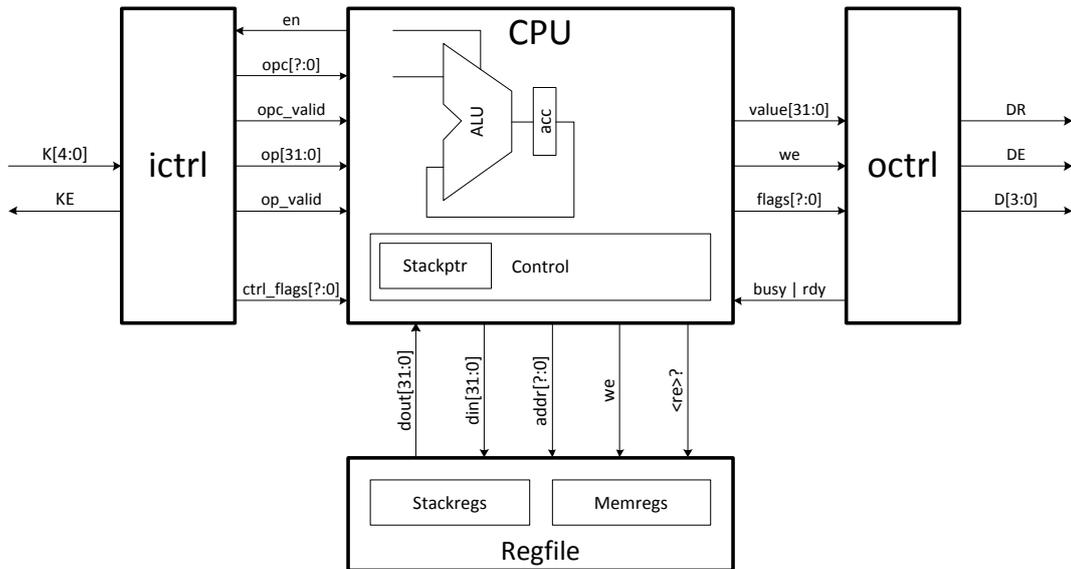


Abbildung 2: Blockschaltbild

- Implementieren Sie das Eingabemodul, das die Eingaben in Binärzahlen umrechnet sowie die auszuführende Operation speichert. Auch sollte das Eingabemodul die eingegebenen Digits zwischenspeichern, so dass das Löschen der zuletzt eingegebenen Ziffer leicht möglich ist. Wird eine Zahl eingegeben, so werden die bisherigen Digits in das nächst höhere Digitregister verschoben und gleichzeitig der Operand (op) neu berechnet und an die CPU weitergereicht, welche die 32-Bit-Zahl an das Anzeigemodul weiterreicht.
- Implementieren Sie das Modul zur Speicherung der Operanden und Ergebnisse, so dass bis zu 8 Zwischenergebnisse und 8 Ergebnisse gespeichert werden können. Diese Funktionen werden später benötigt. Das Speichermodul soll grundsätzlich über ein Speicherinterface angesprochen werden, wie es im Blockschaltbild dargestellt ist.
- Implementieren Sie die Zentrale Einheit mit einer ALU die Zugriff auf das Speichermodul und einen Akkumulator hat.
- Implementieren Sie das Anzeigemodul mit Hilfe des Binär-zu-BCD-Konvertieralgorithmus.
- Erweitern Sie das Design, um die Klammerung und Operationspriorisierung .