Chipentwurf
Übungsblatt 3

4. Januar 2019

1. 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 und das Speichern und Laden 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.

Codierung	Beschreibung
0x00	keine Eingabe
0x01	Negation des Operanden
0x02 - 0x05	Operationen '+', '-', '*', '/'
0x06	Ergebnis berechnen '='
0x07	Speichern des Ergebnisses (gefolgt von einer Zahl von 1-8)
0x08	Laden des Ergebnisses (gefolgt von einer Zahl von 1-8)
0x09	Löschen der letzten Stelle
0x0a	Löschen der Eingabe
0x10 - 0x19	Eingabe der Zahlen 0-9
0x1f	Zurücksetzen

   - Die Ausgabe erfolgt Digitweise, d.h., es wird ein ganzes Zeichen über einen 4-Bit-Datenbus übertragen. Damit lässt sich später beispielsweise eine Anzeige aus 11 Siebensegmentanzeigen realisieren, die über ein Schieberegister angesprochen werden. Entsprechend wird ein Taktsignal DE benötigt:

Pin	Beschreibung
DE	Übernahme von D und Linksshift
D[3:0]	Kodierung der Zahlen und des Negationszeichens

Die Codierung der Zeichen ist wie folgt definiert:

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

Die Ausgabe erfolgt immer als Sequenz von 11 Stellen, wobei das hochwertigste Digit zuerst ausgegeben wird. Belegt die Zahl nicht alle 11 Stellen, werden die führenden Stellen mit Blanks ' ' aufgefüllt. Abbildung 1 zeigt die entsprechende Ausgabesequenz:

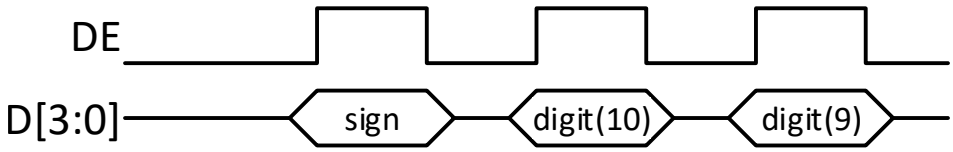

Abbildung 1: Ausgabesequenz

- 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.

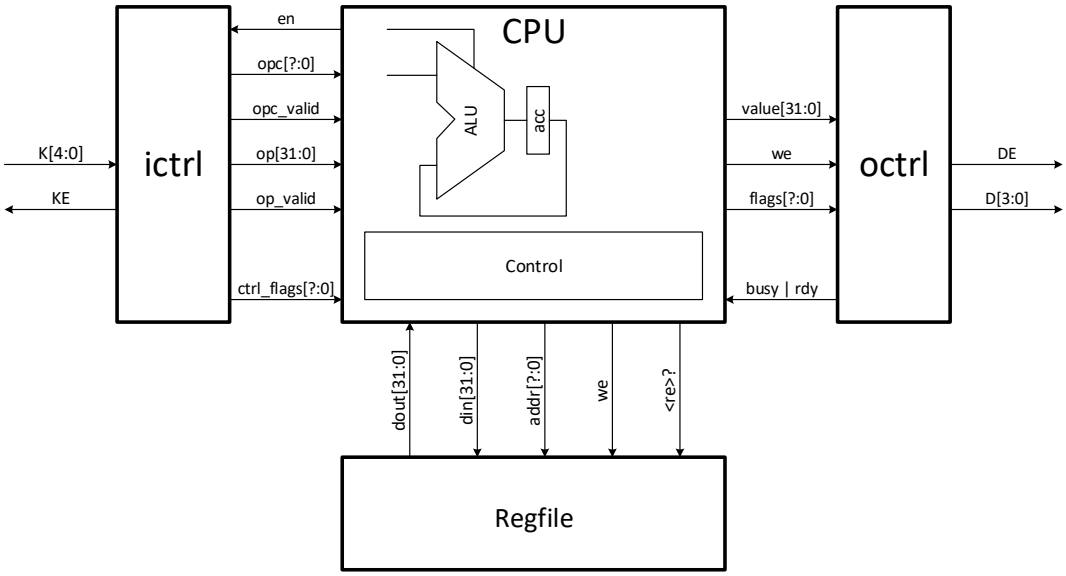

Abbildung 2: Blockschaltbild

- Implementieren Sie das Eingabemodul, dass die Eingaben in Binärzahlen umrechnet, sowie die auszuführende Operation speichert. Auch sollte das Eingabemodul

die eingegebenen Digits zwischenspeichern, so dass das Löschen der zuletzt eingegeben Ziffer leicht möglich ist. Wird eine Zahl eingegeben, so werden die bisherigen Digits in das nächst höhere Digitregister verschoben, gleichzeitig der Operand (op) neu berechnet und an die CPU weitergereicht. Diese wiederum reicht die 32-Bit-Zahl an das Anzeigemodul weiter.

- 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 angedeutet ist.
- Implementieren Sie die Zentrale Einheit mit einer ALU die Zugriff auf das Speichermodul und einen Akkumulator hat. Realisieren Sie den Dividierer als Schaltwerk. Teil der Zentralen Einheit ist auch das Steuerwerk (Control), das die anderen Komponenten überwacht und entsprechende Steuersignale generiert.
- Implementieren Sie das Anzeigemodul mit Hilfe des Binär-zu-BCD-Konvertierungsalgorithmus. Das Komma soll vor der ersten anzuzeigenden Ziffer platzieren werden.