EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

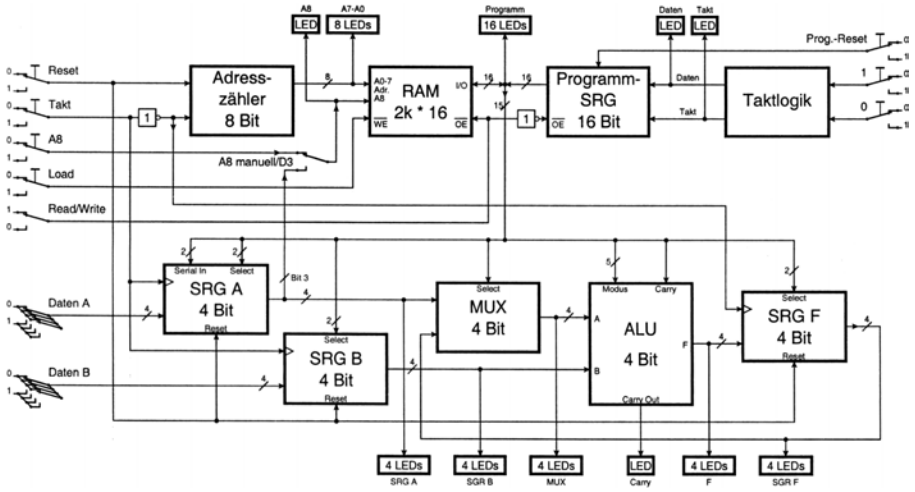
Technische Informatik

Basispraktikum Sommersemester 2001

Protokoll zum Versuchstag 6
Datum: 5.7.2001
Gruppe: David Eißler/ Marc-Oliver Pahl
Autor: Marc-Oliver Pahl

Verwendete Messgeräte:

- Minirechner (MR1)
- Netzgerät (NG2)



Das Programmieren des Minirechners erfolgt durch 15Bit-Worte (-> Tabellen).
Diese werden über das 16Bit-Schieberegister eingegeben (mithilfe der Taster 0 bzw. 1).
Anschließend werden sie im RAM gespeichert und zwar an der Adresse, auf die der Takt gerade zeigt. Dazu muss der Schalter Read/ Write auf Write stehen und dann die Load-Taste gedrückt werden.
Der Reset-Knopf auf der linken Seite schaltet im wesentlichen den Takt zurück auf Null.
Derjenige auf der rechten Seite löscht das 16Bit-Eingabe-SRG.
Um das Programm ablaufen zu lassen, wird der Zähler resettet und der Kippschalter auf Read gestellt. Durch drücken auf die Taste Takt wird der im RAM an der aktuellen Adresse gespeicherte Befehl ausgeführt und der Takt=Adresszähler eins weiter auf den nächsten Befehl im RAM.
Sämtliche relevanten Informationen (nächster Befehl (15Bit) sowie Zustände der Register und Flags) werden durch LEDs angezeigt.

Versuch 1: Ein und Ausgabe auf das RAM testen

a) Dazu haben wir ein paar Binärcodes ins RAM geschrieben, den Zähler resettet und sie dann wieder ausgelesen.

b) Die Eingabewerte kommen über die DIP-Switches...

Befehl					Adresse										Bit	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRG B	SRG A	MUX	ALU	SRG F											IC	SRG B		SRG A				MUX	SRG F		ALU					
					8	7	6	5	4	3	2	1	0		Pin	S1	S0	SIL	SIR	S1	S0	SEL	S1	S0	¬Cn	M	S3	S2	S1	S0
	LD5												0							I	I									
	SHR0												I					0		I	0									
	SHR1											I	0					I		I	0									
	SHL0											I	I						0	0	I									
	SHL1										I	0	0						I	0	I									

c)

Befehl					Adresse										Bit	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRG B	SRG A	MUX	ALU	SRG F											IC	SRG B		SRG A				MUX	SRG F		ALU					
					8	7	6	5	4	3	2	1	0		Pin	S1	S0	SIL	SIR	S1	S0	SEL	S1	S0	¬Cn	M	S3	S2	S1	S0
LD3	LD5	A	A⊕B										0			I	I			I	I	0	I	I	I	I	0	I	I	0
		A	A+B										I									0	I	I	I	0	I	0	0	I
		A	A-B									I	0									0	I	I	0	0	0	I	I	0
		A	(A∨¬B)+(A∧B)+1									I	I									0	I	I	0	0	I	0	I	0
		A	A								I	0	0									0	I	I	I	I	I	I	I	I
		A	A∨B								I	0	I									0	I	I	I	I	I	I	I	0

Versuch 2: Rechnen

a) (15 - 3) / 4

Befehl					Adresse									Bit	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRG B	SRG A	MUX	ALU	SRG F										IC	SRG B		SRG A				MUX	SRG F		ALU					
					8	7	6	5	4	3	2	1	0	Pin	S1	S0	SIL	SIR	S1	S0	SEL	S1	S0	¬Cn	M	S3	S2	S1	S0
LD3	LD15	A	A-B	LD									0		I	I			I	I	0	I	I	0	0	0	I	I	0
				SHR									I									I	0						
				SHR								I	0									I	0						

b) (6 * 2 - 9) * 2

Befehl					Adresse									Bit	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRG B	SRG A	MUX	ALU	SRG F										IC	SRG B		SRG A				MUX	SRG F		ALU					
					8	7	6	5	4	3	2	1	0	Pin	S1	S0	SIL	SIR	S1	S0	SEL	S1	S0	¬Cn	M	S3	S2	S1	S0
	LD6												0						I	I									
	SHL0												I					0	0	I									
LD9		A	A-B	LD								I	0		I	I					0	I	I	0	0	0	I	I	0
				SHL								I	I									0	I						

c) ((4 * 2 - 7) * 2 + 6) / 4

Befehl					Adresse									Bit	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRG B	SRG A	MUX	ALU	SRG F										IC	SRG B		SRG A				MUX	SRG F		ALU					
					8	7	6	5	4	3	2	1	0	Pin	S1	S0	SIL	SIR	S1	S0	SEL	S1	S0	¬Cn	M	S3	S2	S1	S0
	LD4												0						I	I									
	SHL												I					0	0	I									
LD7		A	A-B	LD								I	0		I	I					0	I	I	0	0	0	I	I	0
				SHL								I	I									0	I						
LD6		F	A+B	LD							I	0	0		I	I					I	I	I	I	0	I	0	0	I
				SHR							I	0	I									I	0						
				SHR							I	I	0									I	0						

d) (5 - (1+ 5) / 2) * 4

Befehl					Adresse									Bit	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRG B	SRG A	MUX	ALU	SRG F										IC	SRG B		SRG A				MUX	SRG F		ALU					
					8	7	6	5	4	3	2	1	0	Pin	S1	S0	SIL	SIR	S1	S0	SEL	S1	S0	¬Cn	M	S3	S2	S1	S0
LD5	LD1	A	A+B	LD									0		I	I			I	I	0	I	I	I	0	I	0	0	I
				SHR									I									I	0						
		F	¬A	LD								I	0								I	I	I	I	I	0	0	0	0
		F	A+B+1	LD								I	I								I	I	I	0	0	I	0	0	I
				SHL							I	0	0									0	I						
				SHL							I	0	I									0	I						

Es gibt keinen ALU-Befehl B - A, deshalb wird die Subtraktion durch Addition des Zweierkomplements (=Einerkomplement+1) ersetzt...

Versuch 3: Division durch 7

Unser Algorithmus, der die richtigen Ergebnisse für die ganzzahlige Division im Bereich 0-15 abdeckt ist 1r(2r(A) + 1).

Die Funktionstabelle für die damit gültigen Werte ist:

Eingabe (A)	Ausgabe (1r(2r(A) + 1))	
I I I 0	0 0 I 0	(14/7 = 2)
0 I I I	0 0 0 I	(7/7 = 1)
0 0 0 0	0 0 0 0	(0/7 = 0)

Befehl					Adresse									Bit	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRG B	SRG A	MUX	ALU	SRG F										IC	SRG B		SRG A				MUX	SRG F		ALU					
					8	7	6	5	4	3	2	1	0	Pin	S1	S0	SIL	SIR	S1	S0	SEL	S1	S0	¬Cn	M	S3	S2	S1	S0
	LD14	A	A	LD									0						I	I	0	I	I	I	I	I	I	I	I
				SHR									I									I	0						
				SHR								I	0									I	0						
		F	A+1	LD								I	I								I	I	I	I	0	I	0	0	I
				SHR							I	0	0									I	0						

Versuch 4: Multiplikation zweier 4Bit-Zahlen

Der Algorithmus funktioniert fast so, wie man ihn auch von Hand auf dem Blatt durchführen würde, nur werden die Stelligkeitsverschiebungen erst im Nachhinein aufgebaut.

```
I I 0 0 * I 0 I 0
------------------
I I 0 0 0 0 0
+   I I 0 0 0
------------------
 I I I  I 0 0 0
```

Befehl					Adresse									Bit	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRG B	SRG A	MUX	ALU	SRG F										IC	SRG B		SRG A				MUX	SRG F		ALU					
					8	7	6	5	4	3	2	1	0	Pin	S1	S0	SIL	SIR	S1	S0	SEL	S1	S0	¬Cn	M	S3	S2	S1	S0
LD	LD												0		I	I			I	I									
	SHL			NOP									I					0	0	I									
	SHL		B	LD	I								I					0	0	I		I	I	I	I	I	0	I	0
				SHL								I	0									0	I						
				SHL	I							I	0									0	I						
	SHL			NOP								I	I					0	0	I									
	SHL	F	A+B	LD	I							I	I					0	0	I		I	I	I	0	I	0	0	I
				SHL							I	0	0									0	I						
				SHL	I						I	0	0									0	I						
	SHL			NOP							I	0	I					0	0	I									
	SHL	F	A+B	LD	I						I	0	I					0	0	I		I	I	I	0	I	0	0	I
				SHL							I	I	0									0	I						
				SHL	I						I	I	0									0	I						
				NOP							I	I	I																
		F	A+B	LD	I						I	I	I									I	I	I	0	I	0	0	I

Der Algorithmus „erstellt“ die roten Nullen erst im Nachhinein durch das nach links Shiften in jedem Schritt...