



**h\_da**

HOCHSCHULE DARMSTADT  
UNIVERSITY OF APPLIED SCIENCES

**fbi**  
FACHBEREICH INFORMATIK

RECHNERARCHITEKTUR

WS2023

**Termin 2**

Umgang Befehlssatz eines MU1 Prozessors

Name, Vorname	Matrikelnummer	Anmerkungen
Datum	Raster (z.B. Mi3x)	Testat/Datum

Legende: V:Vorbereitung, D: Durchführung, P: Protokoll/Dokumentation, T: Testat

## Vorbereitung

Bereiten Sie die Lösungen daheim oder in den offenen Laboren so vor, dass Sie die Ergebnisse zum Labortermin präsentieren können.

### Aufgabe1:

Erweitern sie den Befehlssatz des MU1 Prozessors um die Befehle PUSH, POP, LDR S, STR S, MOV PC und MOV SP. Zeichnen Sie in die Diagramme den jeweiligen Datenfluss und füllen Sie die Steuerungstabelle aus.

Der Befehl PUSH dekrementiert ( $SP=SP-1$ ) den Stackpointer (Register SP) und speichert den aktuellen Akkumulatorinhalt (Register A) auf dem Stack.

Der Befehl POP lädt den Wert auf den der Stackpointer zeigt in den Akkumulator und inkrementiert ( $SP=SP+1$ ) den Stackpointer.

Der Befehl STR S schreibt den Inhalt des Akkumulator in die Speicherstelle mit der Adresse, welche in der Speicherstelle mit der Adresse S steht.

Der Befehl LDR S lädt den Inhalt der Speicherstelle mit der Adresse, welche in der Speicherstelle mit der Adresse S steht, in den Akkumulator.

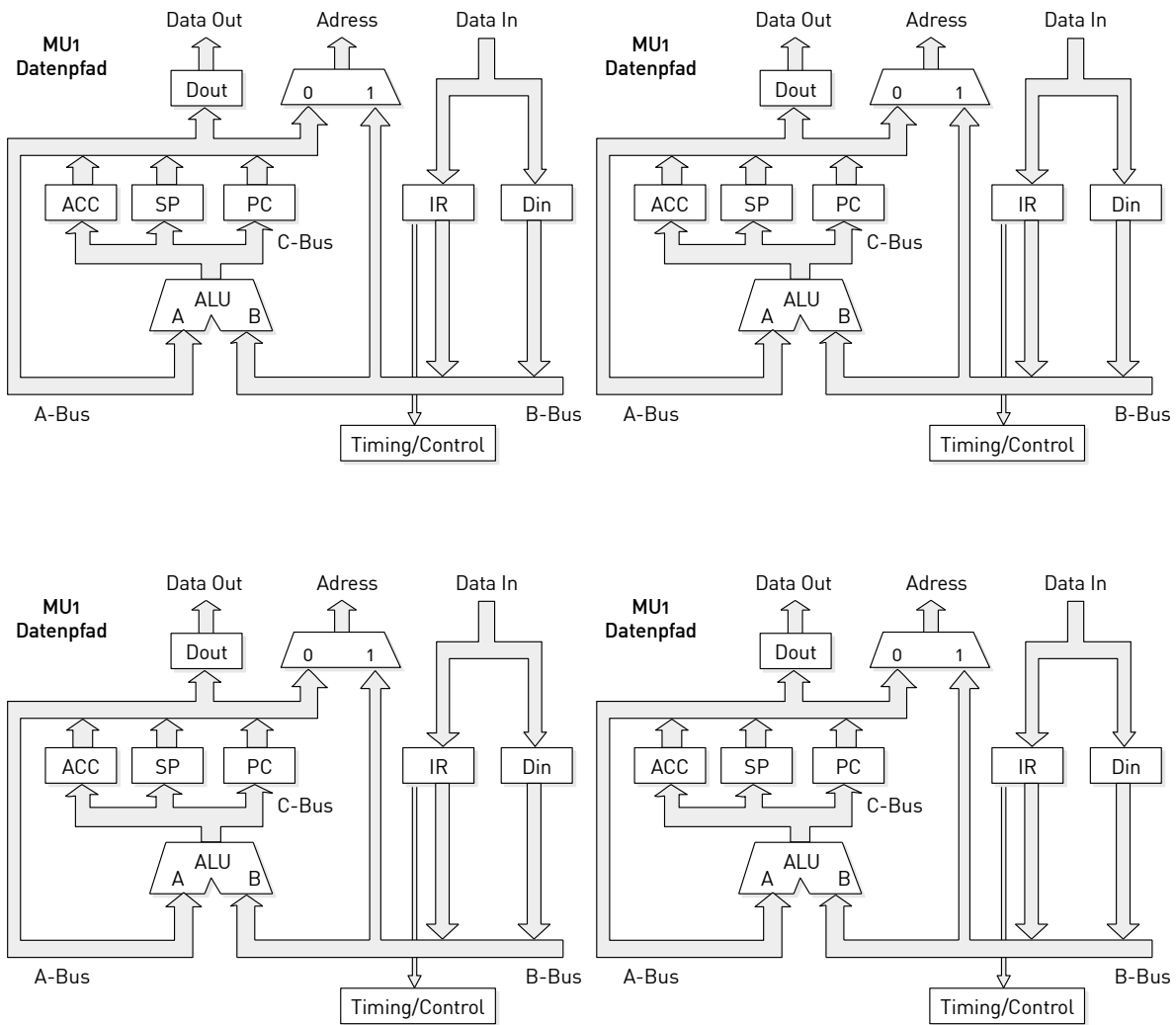
Der Befehl MOV PC kopiert den Inhalt vom Register ACC in das Register PC.

Der Befehl MOV SP kopiert den Inhalt vom Register ACC in das Register SP.

Befehlstabelle für MU1

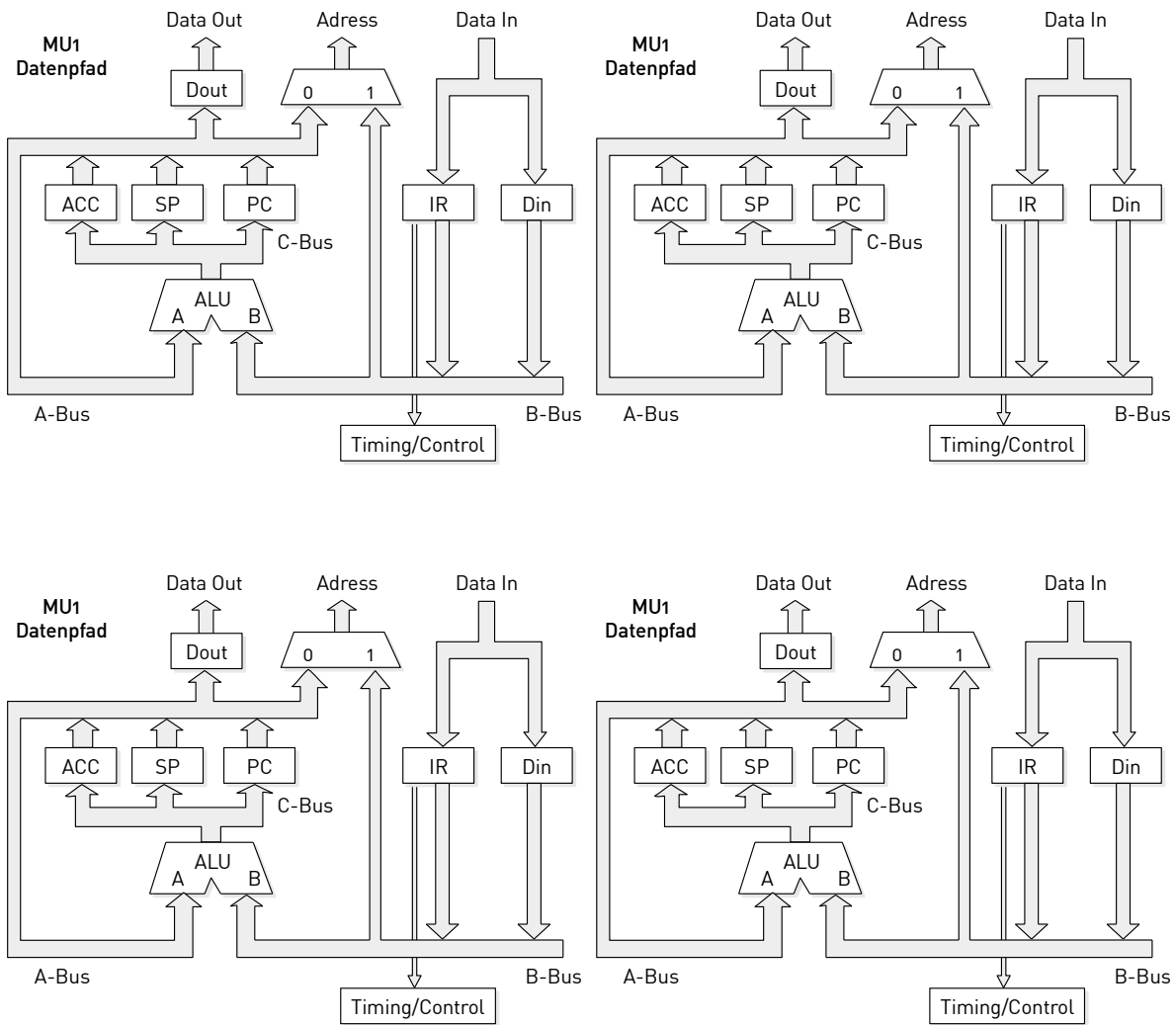
<i>Instruction</i>	<i>Effekt</i>
Reset	PC = 0
LDA S	ACC = [S]
STO S	[S] = ACC
ADD S	ACC = ACC + [S]
JUMP S	PC = S
JGE S	IF ACC >= 0 PC = S
JNE S	IF ACC != 0 PC = S
STOP	stop
CALL S	SP = SP-1, [SP] = PC, PC = S
RETURN	PC = [SP], SP = SP + 1
PUSH	SP = SP-1, [SP] = ACC
POP	ACC = [SP], SP = SP + 1
LDR S	ACC = [[S]]
STR S	[[S]] = ACC
MOV PC	PC = ACC
MOV SP	SP = ACC

## Der Befehl Push



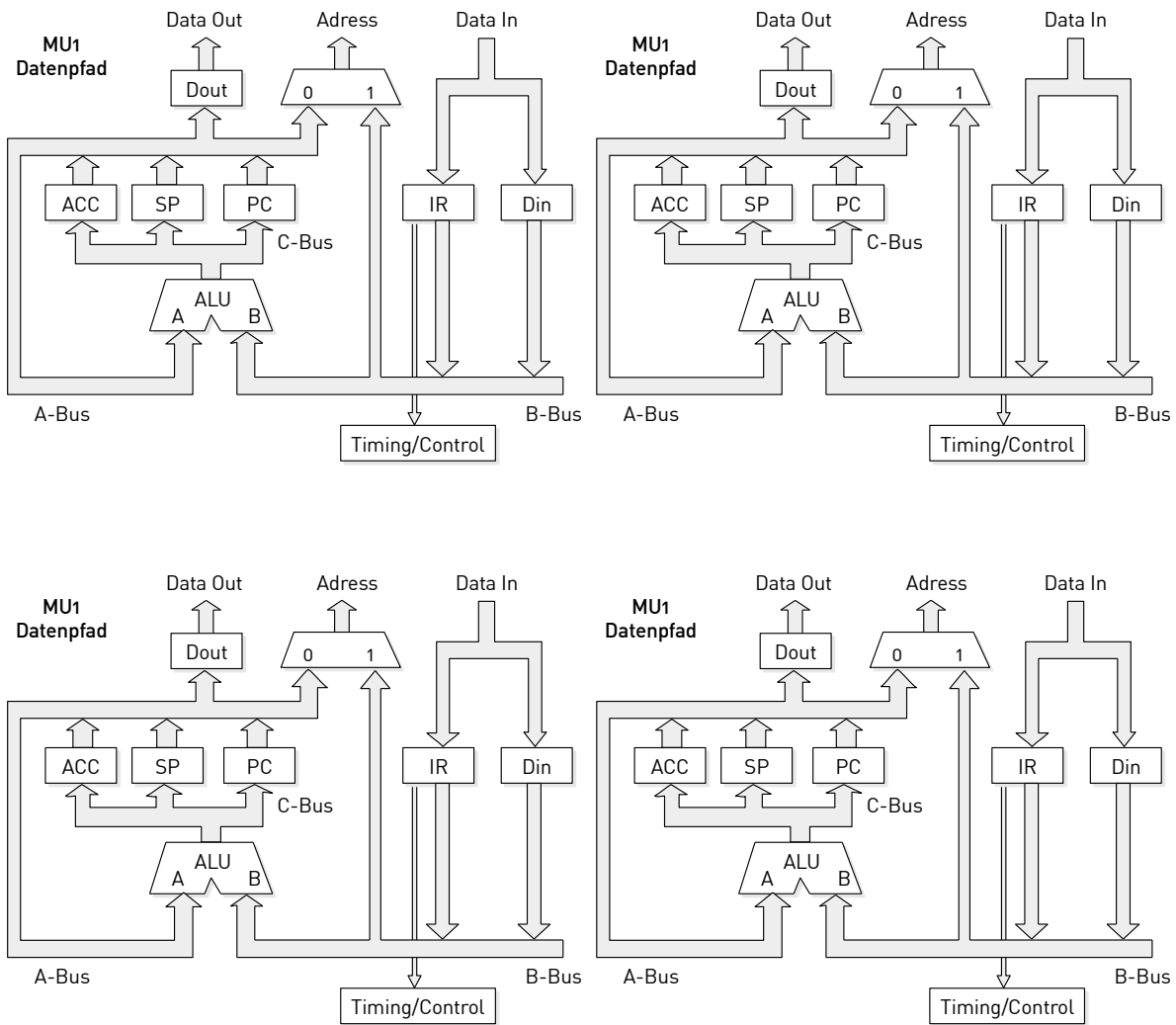
Inputs		Outputs															Description/Effect								
Instruction	Opcode	Reset	Step	ACC <sub>Z</sub> /Zero	ACC <sub>15</sub> /Negativ	Step	ACC <sub>0E</sub>	ACC <sub>1E</sub>	SP <sub>oe</sub>	SP <sub>ie</sub>	PC <sub>oe</sub>	PC <sub>ie</sub>	IR <sub>oe</sub>	IR <sub>ie</sub>	DIN <sub>oe</sub>	DIN <sub>ie</sub>	DOUT <sub>oe</sub>	DOUT <sub>ie</sub>	ALU	Address	Memory				
		MEM <sub>rq</sub>	R/W																						
<b>PUSH</b>	<b>1010<sub>b</sub> = 0xA</b>																								

## Der Befehl Pop



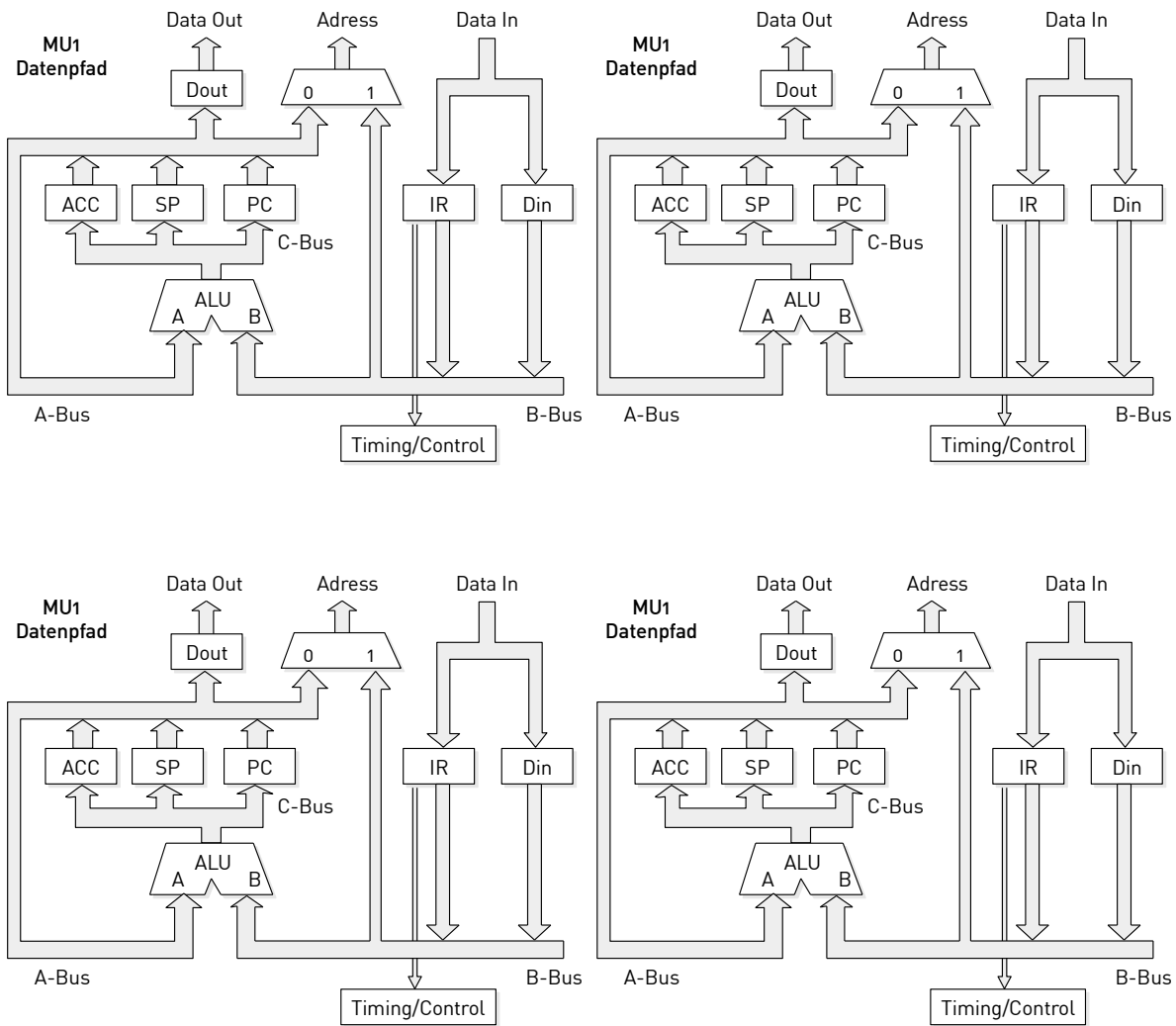
Inputs		Outputs														Description/Effect							
Instruction	Opcode	Reset	Step	ACC <sub>Z</sub> / Zero	ACC <sub>15</sub> / Negativ	Step	ACC <sub>oe</sub>	ACC <sub>ie</sub>	SP <sub>oe</sub>	SP <sub>ie</sub>	PC <sub>oe</sub>	PC <sub>ie</sub>	IR <sub>oe</sub>	IR <sub>ie</sub>	DIN <sub>oe</sub>	DIN <sub>ie</sub>	DOUT <sub>oe</sub>	DOUT <sub>ie</sub>	ALU	Memory			
																				Address	MEM <sub>rq</sub>	R/W	
POP																							

## Der LDR S Befehl



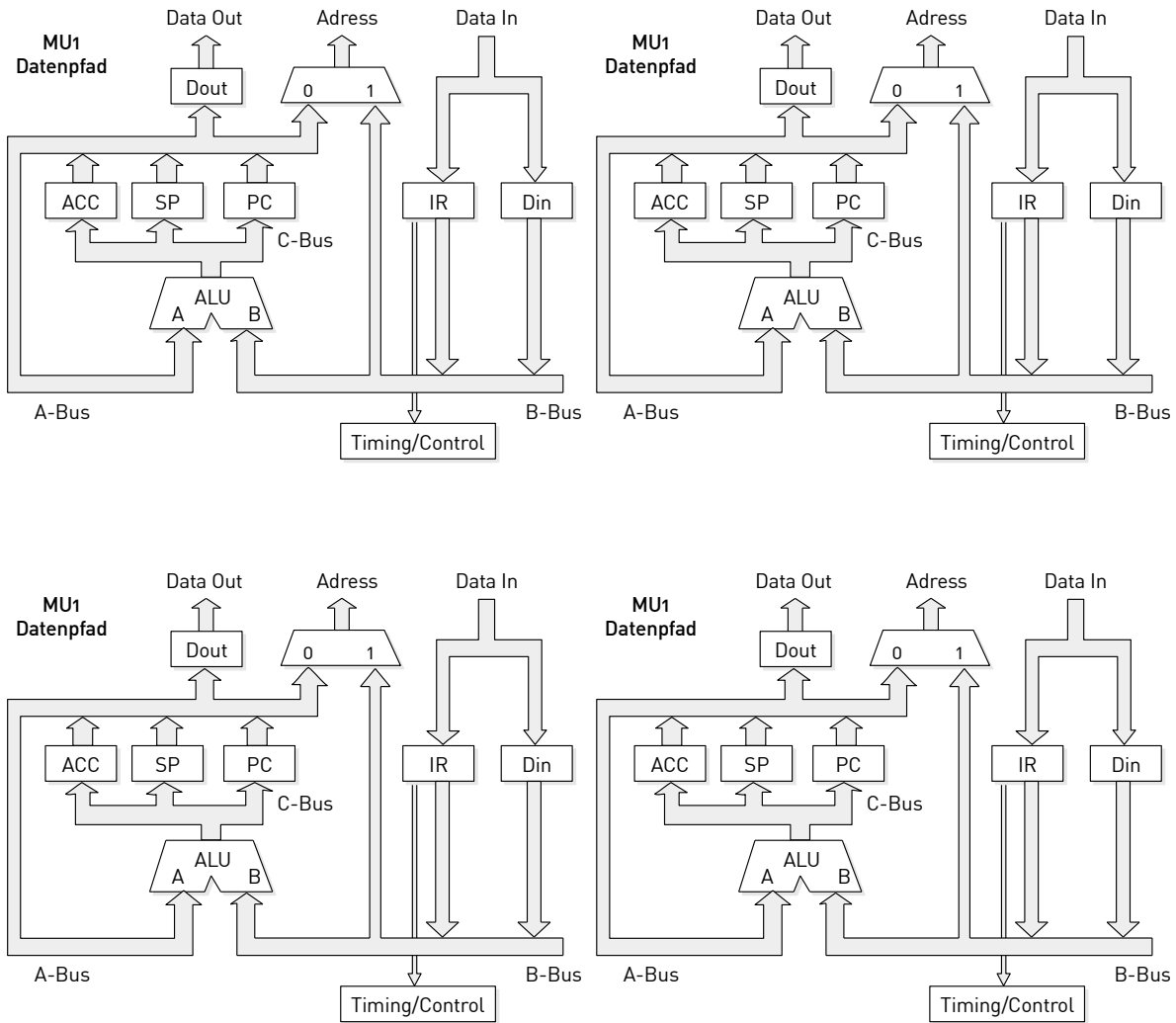
Inputs		Outputs															Description/Effect								
Instruction	Opcode	Reset	Step	ACC <sub>Z</sub> /Zero	ACC <sub>15</sub> /Negativ	Step	ACC <sub>oe</sub>	ACC <sub>ie</sub>	SP <sub>oe</sub>	SP <sub>ie</sub>	PC <sub>oe</sub>	PC <sub>ie</sub>	IR <sub>oe</sub>	IR <sub>ie</sub>	DIN <sub>oe</sub>	DIN <sub>ie</sub>	DOUT <sub>oe</sub>	DOUT <sub>ie</sub>	ALU	Memory					
																				Address	MEM <sub>rq</sub>	R/W			
LDR S																									

## Der STR S Befehl



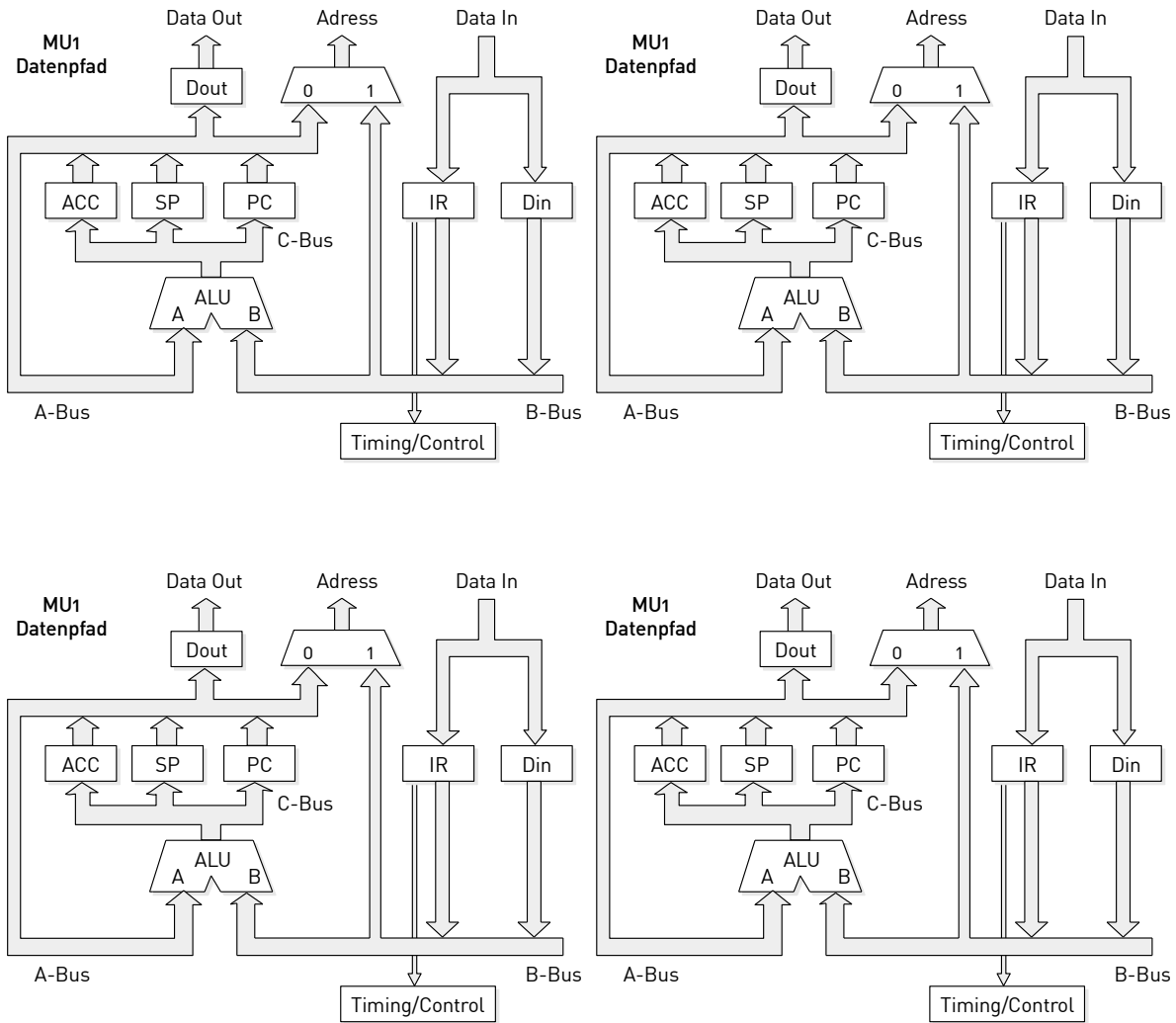
Inputs		Outputs															Description/Effect							
Instruction	Opcode	Reset	Step	ACC <sub>Z</sub> / Zero	ACC <sub>15</sub> / Negativ	Step	ACC <sub>0E</sub>	ACC <sub>1E</sub>	SP <sub>0E</sub>	SP <sub>1E</sub>	PC <sub>0E</sub>	PC <sub>1E</sub>	IR <sub>0E</sub>	IR <sub>1E</sub>	DIN <sub>0E</sub>	DIN <sub>1E</sub>	DOUT <sub>0E</sub>	DOUT <sub>1E</sub>	ALU	Memory				
																				Address	MEM <sub>rq</sub>	R/W		
STR S																								

## Der MOV PC Befehl



Inputs		Outputs														Description/Effect								
Instruction	Opcode	Reset	Step	ACC <sub>Z</sub> / Zero	ACC <sub>15</sub> / Negativ	Step	ACC <sub>oe</sub>	ACC <sub>ie</sub>	SP <sub>oe</sub>	SP <sub>ie</sub>	PC <sub>oe</sub>	PC <sub>ie</sub>	IR <sub>oe</sub>	IR <sub>ie</sub>	DIN <sub>oe</sub>	DIN <sub>ie</sub>	DOUT <sub>oe</sub>	DOUT <sub>ie</sub>	ALU	Memory				
																				Address	MEM <sub>rq</sub>	R/W		
<b>MOV PC</b>																								

## Der MOV SP Befehl



Inputs		Outputs															Description/Effect							
Instruction	Opcode	Reset	Step	ACC <sub>z</sub> /Zero	ACC <sub>15</sub> /Negativ	Step	ACC <sub>oe</sub>	ACC <sub>ie</sub>	SP <sub>oe</sub>	SP <sub>ie</sub>	PC <sub>oe</sub>	PC <sub>ie</sub>	IR <sub>oe</sub>	IR <sub>ie</sub>	DIN <sub>oe</sub>	DIN <sub>ie</sub>	DOUT <sub>oe</sub>	DOUT <sub>ie</sub>	ALU	Memory				
																				Address	MEM <sub>rq</sub>	R/W		
<b>MOV SP</b>																								



## Aufgabe2:

Versuchen sie das Beispielprogramm aus der Vorlesung mit den neuen Befehlen LDR S und STR S so umzuschreiben, dass sie keinen selbst modifizierenden Code mehr benötigen.

```
Loop:      LDA   Total      ; Accumulate total
Add_instr: ADD   Table      ; Begin at head of table
           STO   Total      ;
           LDA   Add_instr  ; Change address ...
           ADD   One        ; by modifying instruction!
           STO   Add_instr  ;
           LDA   Count      ; Count iterations
           SUB   One        ; Count down to zero
           STO   Count      ;
           JGE  Loop        ; If >= 0 repeat
           STP                ; Halt execution
```

; Data definitions

```
Total      DEFW  0      ; Total - initially zero
One         DEFW  1      ; The number one
Count DEFW  4      ; Loop counter (loop 5x)
Table      DEFW  39     ; The numbers to total ...
           DEFW  25     ;
           DEFW  4      ;
           DEFW  98     ;
           DEFW  17     ;
```

Diese Aufgabe wird in einem der weiteren Termine (evtl. Termin 5) nochmals behandelt werden.