RECHNERARCHITEKTUR

Termin 2

Umgang Befehlssatz eines MU1 Prozessors

Vorbereitung

Bereiten Sie die Lösungen daheim so vor, dass Sie die Ergebnisse zum Praktikumstermin präsentieren können.

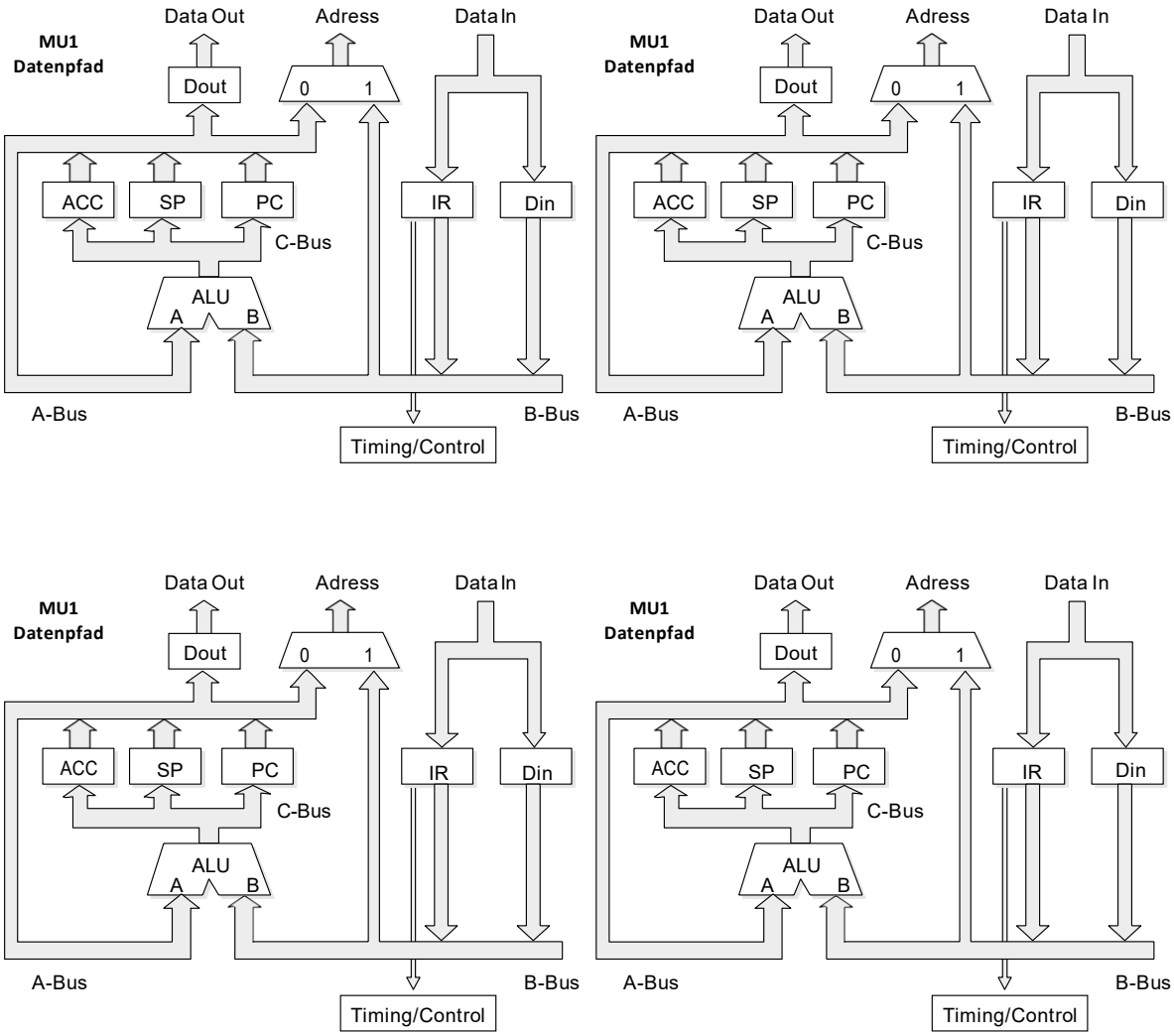
Aufgabe1:

Zeichnen Sie für die untenstehenden Befehle den jeweiligen Datenfluss und füllen Sie die Steuerungstabelle aus.

Befehlstabelle für MU1

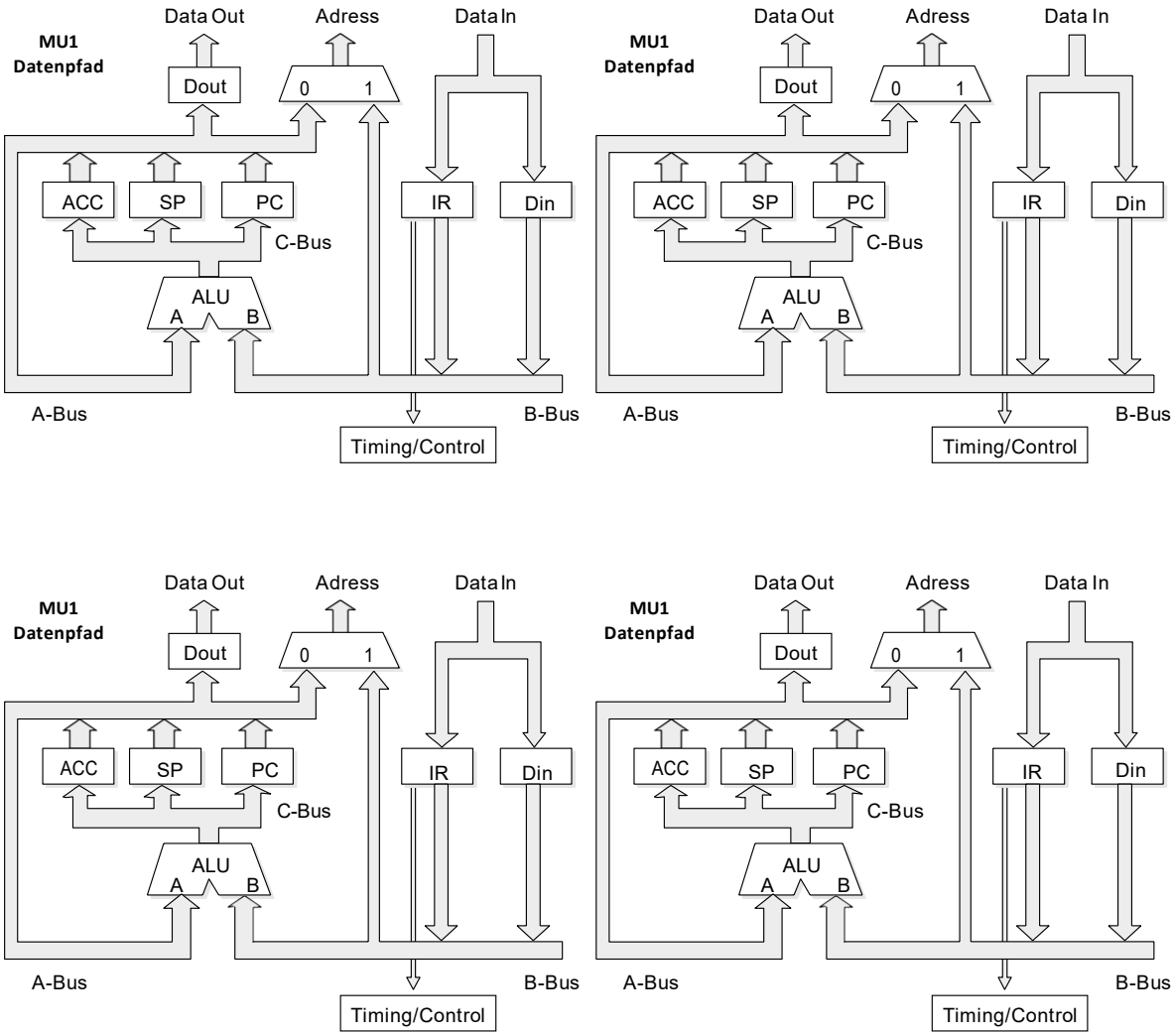
Instruction	***Effekt***
Reset	PC = 0
LDA S	ACC = [S]
STO S	[S] = ACC
ADD S	ACC = ACC + [S]
JUMP S	PC = S
JGE S	IF ACC >= 0 PC = S
JNE S	IF ACC = 0 PC = S
STOP	stop
CALL S	SP = SP-1, [SP] = PC, PC = S
RETURN	PC = [SP], SP = SP + 1
PUSH	SP = SP-1, [SP] = ACC
POP	ACC = [SP], SP = SP + 1
LDR S	ACC = [[S]]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

Der Befehl Push



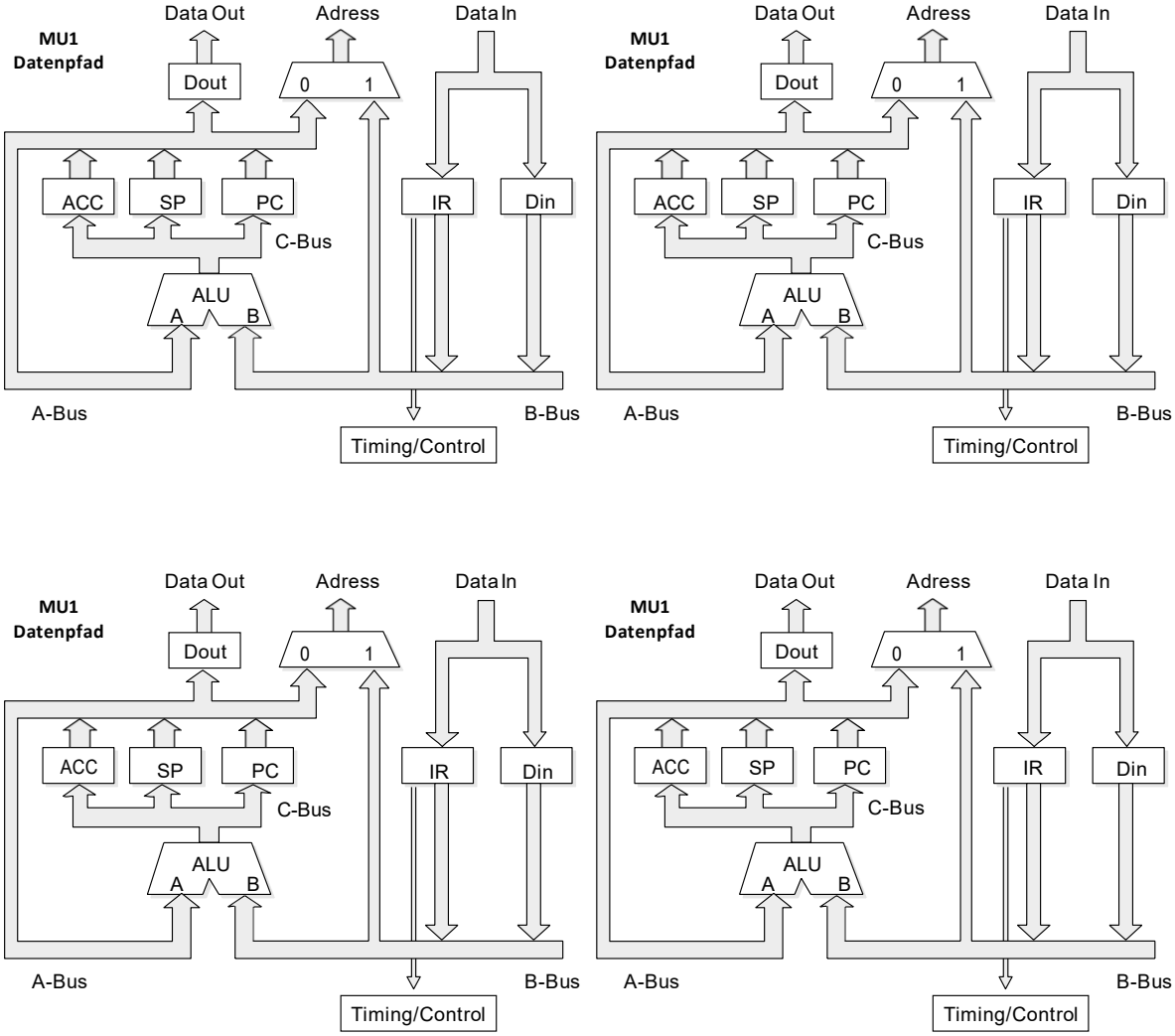
Inputs						Outputs																Description
Instruction	Opcode	/Reset	Step	ACC_Z/ Zero	ACC_{15}/Negativ	Step	Adress	ACC_{OE}	ACC_{ie}	PC_{oe}	PC_{ie}	IR_{oe}	IR_{ie}	SP_{oe}	SP_{ie}	DIN_{oe}	DIN_{ie}	$DOUT_{oe}$	$DOUT_{ie}$	ALU Function	MEM_{rq}	RnW		
PUSH																						

Der Befehl Pop



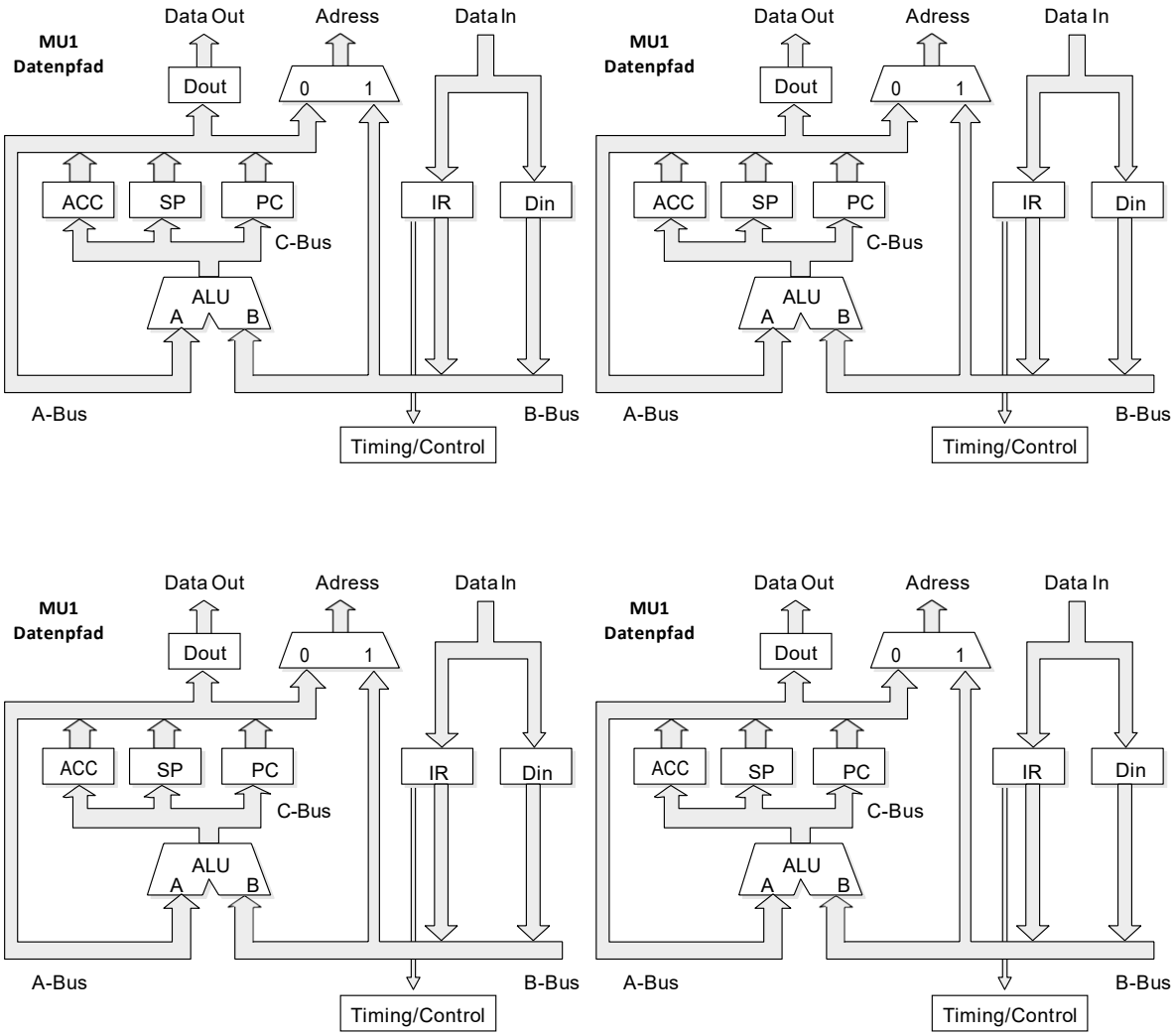
Inputs						Outputs																Description
Instruction	Opcode	/Reset	Step	ACC_Z/ Zero	ACC_{15}/Negativ	Step	Adress	ACC_{OE}	ACC_{ie}	PC_{oe}	PC_{ie}	IR_{oe}	IR_{ie}	SP_{oe}	SP_{ie}	DIN_{oe}	DIN_{ie}	$DOUT_{oe}$	$DOUT_{ie}$	ALU Function	MEM_{rq}	RnW		
POP																						

Der LDR S Befehl



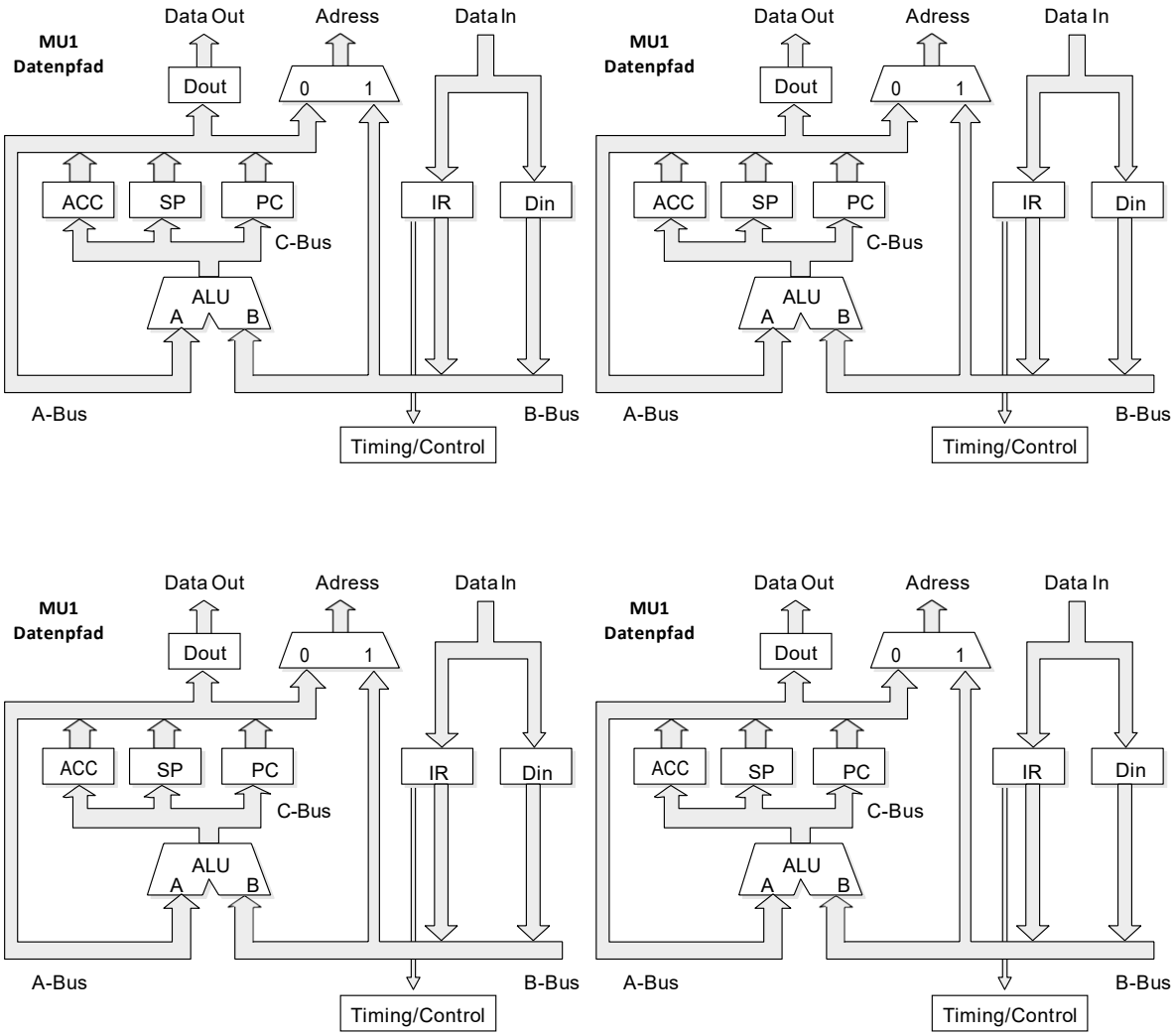
Inputs						Outputs																Description
Instruction	Opcode	/Reset	Step	ACC_Z/ Zero	ACC_{15}/Negativ	Step	Adress	ACC_{OE}	ACC_{ie}	PC_{oe}	PC_{ie}	IR_{oe}	IR_{ie}	SP_{oe}	SP_{ie}	DIN_{oe}	DIN_{ie}	$DOUT_{oe}$	$DOUT_{ie}$	ALU Function	MEM_{rq}	RnW		
LDR S																						

Der STR S Befehl



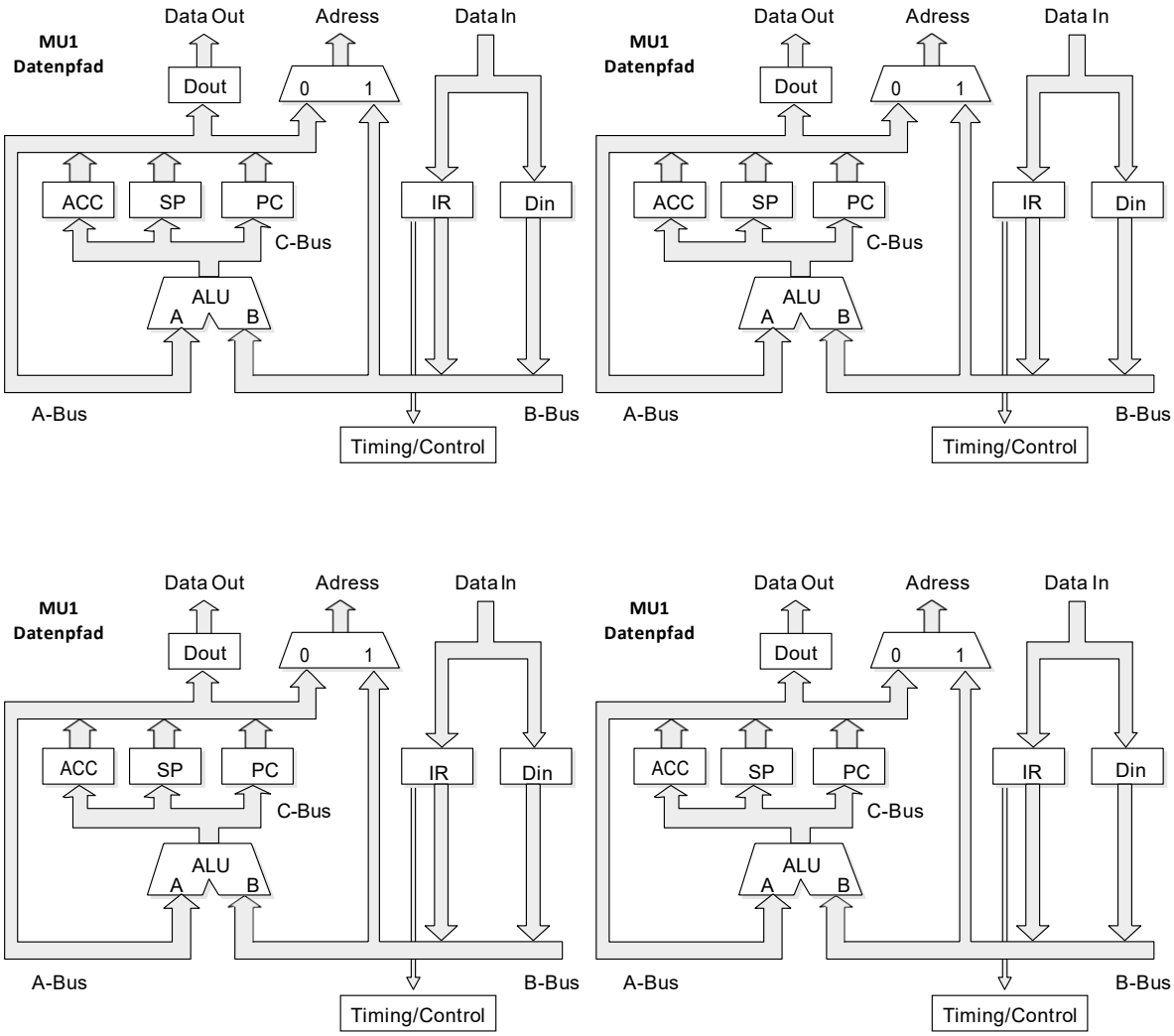
Inputs						Outputs																Description
Instruction	Opcode	/Reset	Step	ACC_Z/ Zero	ACC_{15}/Negativ	Step	Adress	ACC_{OE}	ACC_{ie}	PC_{oe}	PC_{ie}	IR_{oe}	IR_{ie}	SP_{oe}	SP_{ie}	DIN_{oe}	DIN_{ie}	$DOUT_{oe}$	$DOUT_{ie}$	ALU Function	MEM_{rq}	RnW		
STR S																						

Der MOV PC Befehl



Inputs						Outputs																Description
Instruction	Opcode	/Reset	Step	ACC_Z/ Zero	ACC_{15}/Negativ	Step	Adress	ACC_{OE}	ACC_{ie}	PC_{oe}	PC_{ie}	IR_{oe}	IR_{ie}	SP_{oe}	SP_{ie}	DIN_{oe}	DIN_{ie}	$DOUT_{oe}$	$DOUT_{ie}$	ALU Function	MEM_{rq}	RnW	
MOV PC																						

Der MOV SP Befehl



Inputs						Outputs																Description
Instruction	Opcode	/Reset	Step	ACC_Z/ Zero	ACC_{15}/Negativ	Step	Adress	ACC_{OE}	ACC_{ie}	PC_{oe}	PC_{ie}	IR_{oe}	IR_{ie}	SP_{oe}	SP_{ie}	DIN_{oe}	DIN_{ie}	$DOUT_{oe}$	$DOUT_{ie}$	ALU Function	MEM_{rq}	RnW		
MOV SP																						

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       ;
```