



h_da

HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

fbi
FACHBEREICH INFORMATIK

RECHNERARCHITEKTUR

WS2016/17

Termin 2

Umgang Befehlssatz eines MU0-2 Prozessors

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

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

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 MU0-2 Prozessors um die Befehle PUSH, POP, LDR S und STR S .

Malen 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.

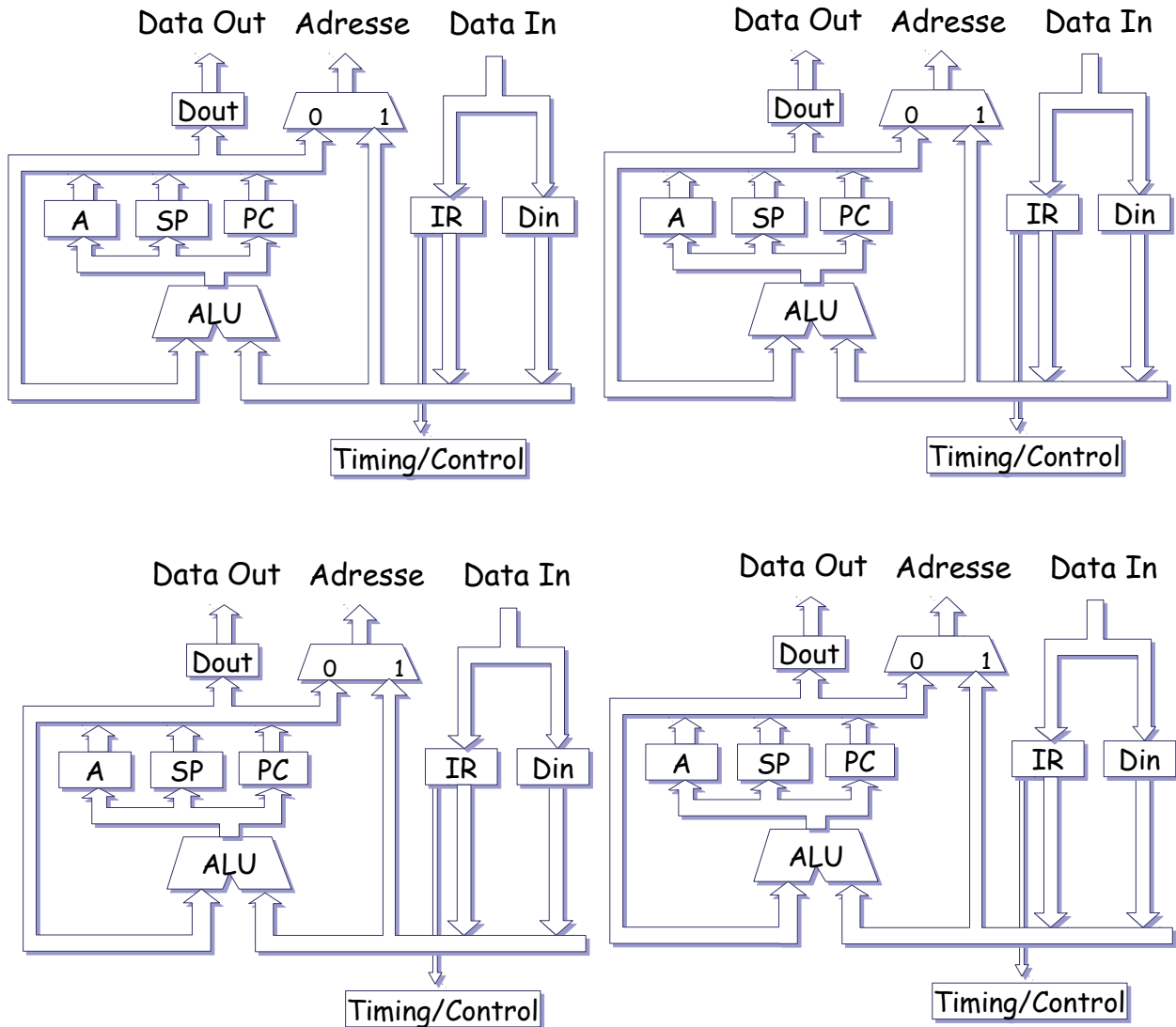
Der Befehl STR S schreibt den Inhalt des Akkumulator in die Speicherstelle mit der Adresse, welche in der Speicherstelle mit der Adresse S steht.

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

Befehlstabelle für MU0-2

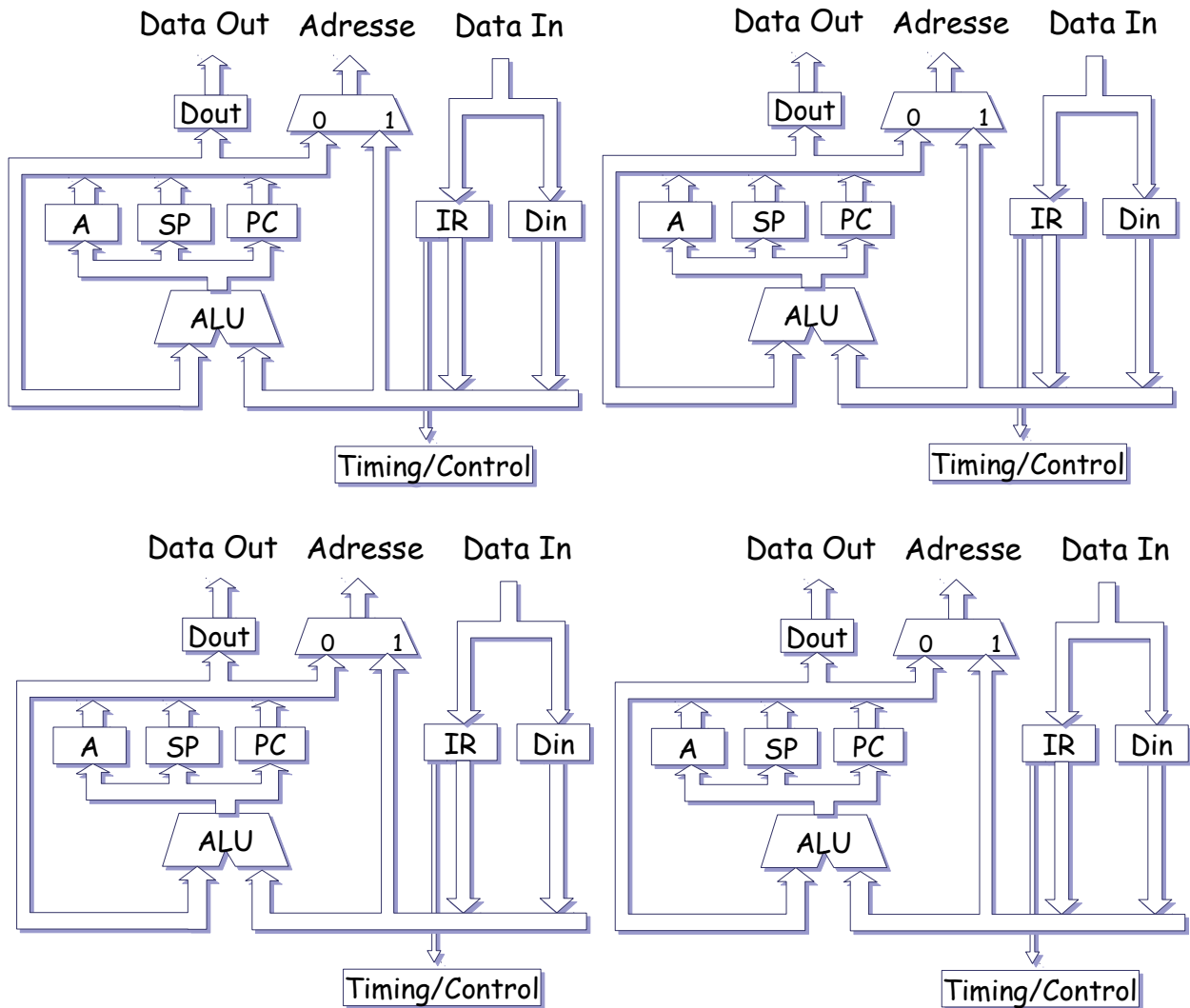
<i>Instruction</i>	<i>Effekt</i>	<i>Instruction</i>	<i>Effekt</i>
Reset	$PC = 0$		
LDA S	$A = [S]$	CALL S	$SP = SP-1, [SP] = PC, PC = S$
STO S	$[S] = A$	RETURN	$PC = [SP], SP = SP + 1$
ADD S	$A = A + [S]$	PUSH	$SP = SP-1, [SP] = A$
SUB S	$A = A - [S]$	POP	$A = [SP], SP = SP + 1$
JMP S	$PC = S$	LDR S	$A = [[S]]$
JGE S	IF $A \geq 0$ $PC = S$	STR S	$[[S]] = A$
JNE S	IF $A \neq 0$ $PC = S$	MOV PC	$PC = A$
STOP	stop	MOV SP	$SP = A$

Der Befehl Push



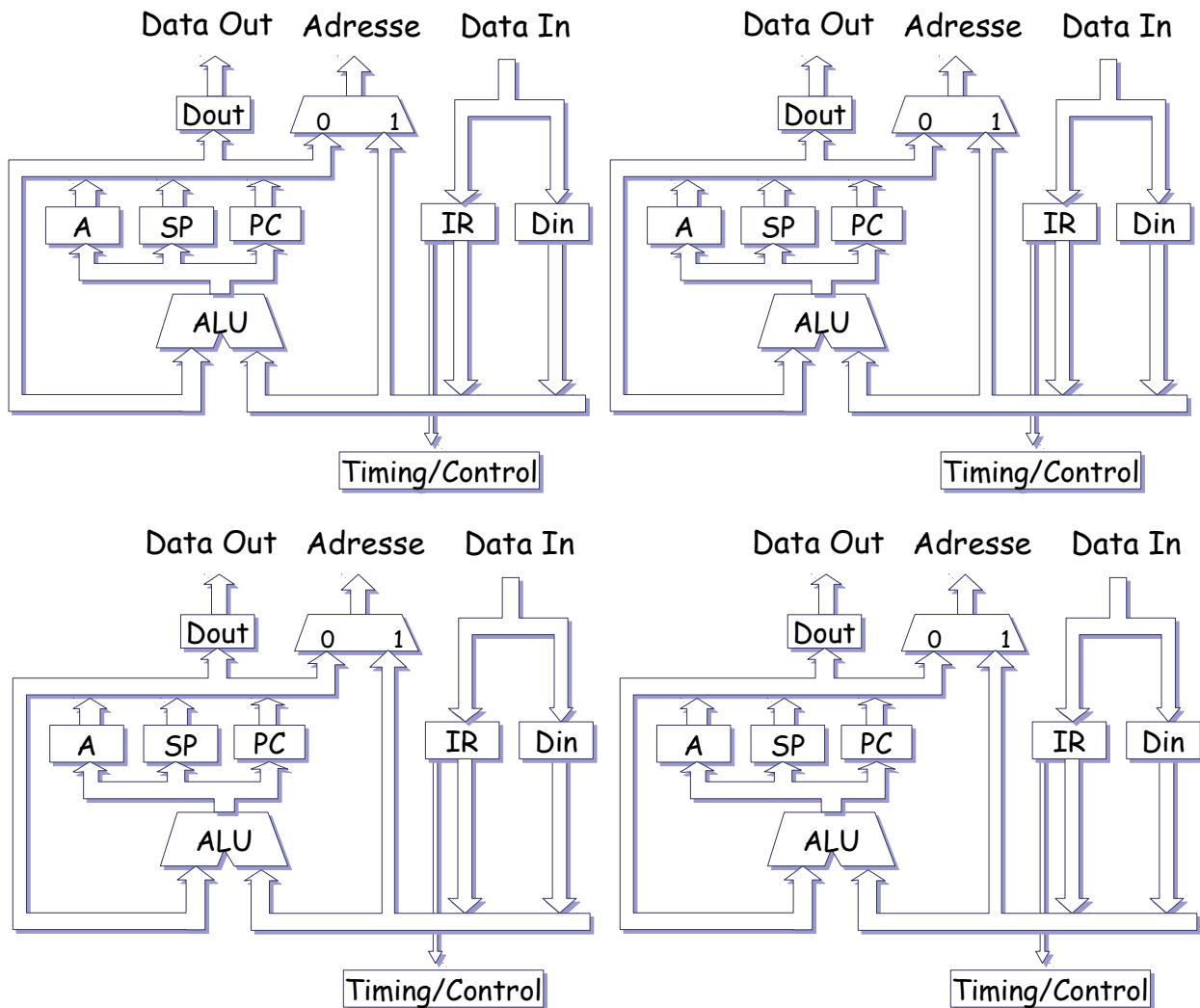
Inputs		Outputs																				
Instruction	Opcode	Reset	step	ACCz	ACC15	step	Address	Aoe	Aie	Pcoe	Pcie	Iroe	Irie	Spoe	Spie	DINoe	DINie	DOUToe	DOUTie	ALU Function	MEMIrq	RnW
PUSH																						

Der Befehl Pop



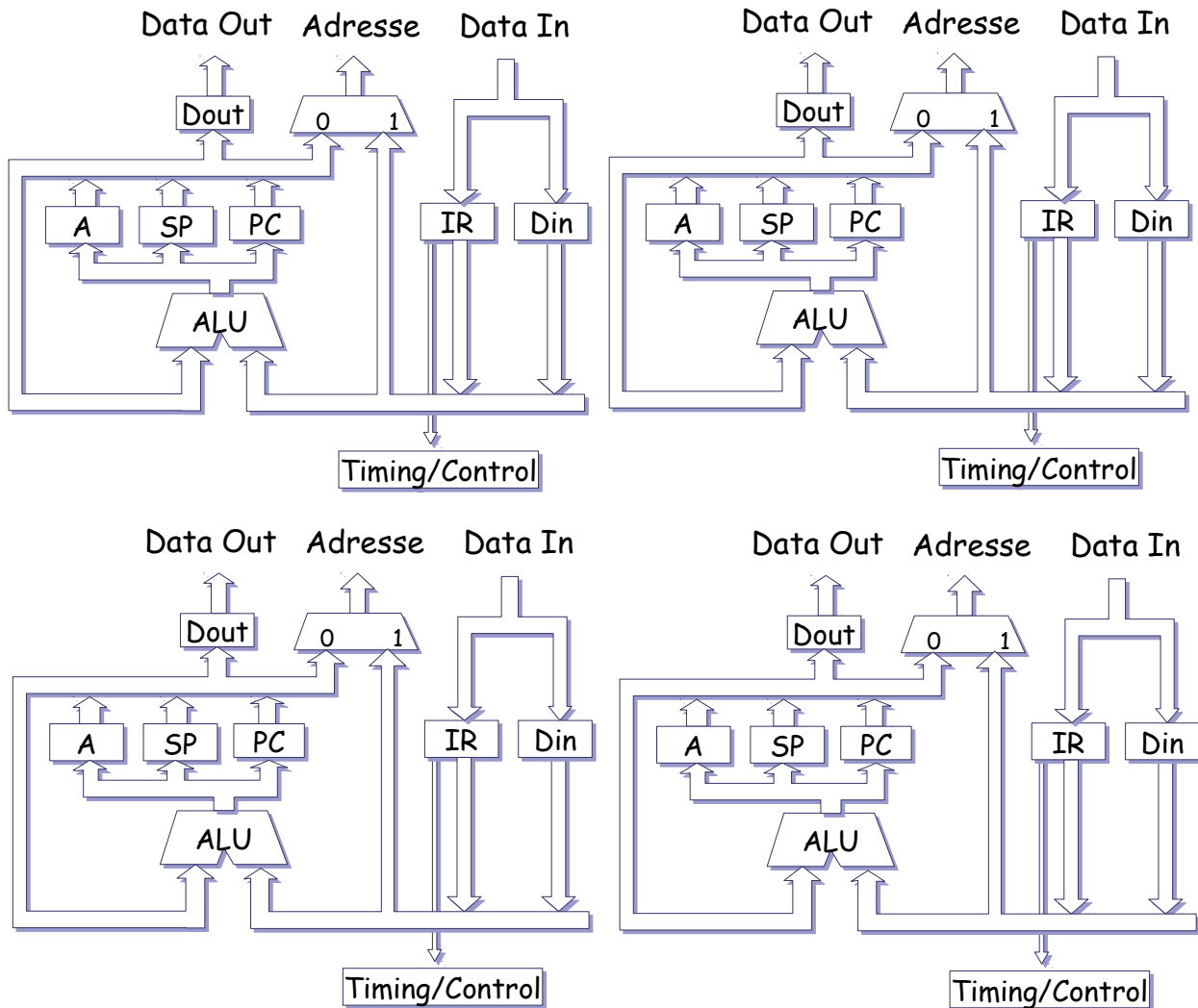
Inputs		Outputs																				
Instruction	Opcode	Reset	step	ACCz	ACC15	step	Address	Aoe	Aie	Pcoe	Pcie	Iroe	Irie	Spoe	Spie	DINoe	DINie	DOUToe	DOUTie	ALU Function	MEMrq	RnW
POP																						

Der STR S Befehl



Inputs		Outputs																				
Instruction	Opcode	Reset	step	ACCz	ACC15	step	Address	Aoe	Aie	Pcoe	Pcie	Iroe	Irie	Spoe	Spie	DINoe	DINie	DOUToe	DOUTie	ALU Function	MEMrq	RnW
STR S																						

Der LDR S Befehl



Inputs		Outputs																				
Instruction	Opcode	Reset	step	ACCz	ACC15	step	Address	Aoe	Aie	Pcoe	Pcie	Iroe	Irie	Spoe	Spie	DINoe	DINie	DOUToe	DOUTie	ALU Function	MEMrq	RnW
LDR S																						

Zusatzaufgabe:

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      ;
```