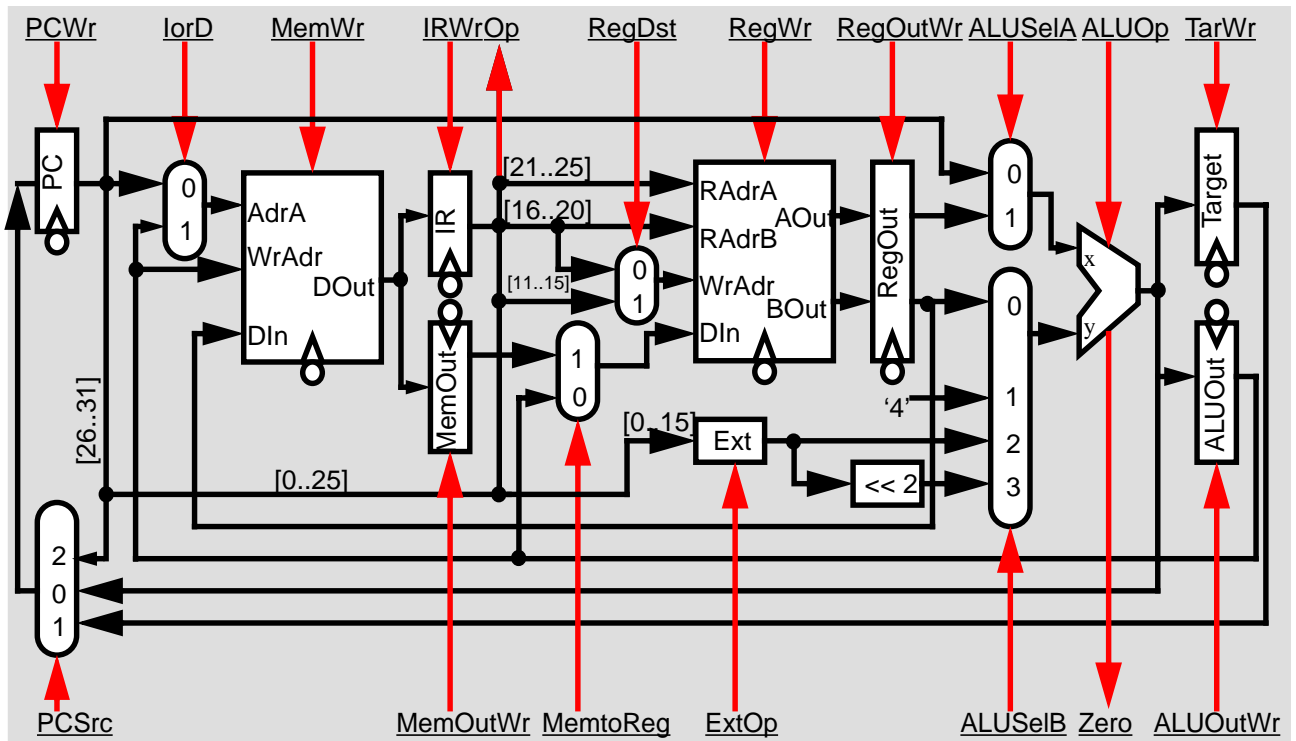


Aufgabe 5: Datenpfad und Mikroprogrammierung (Total 25 Punkte)

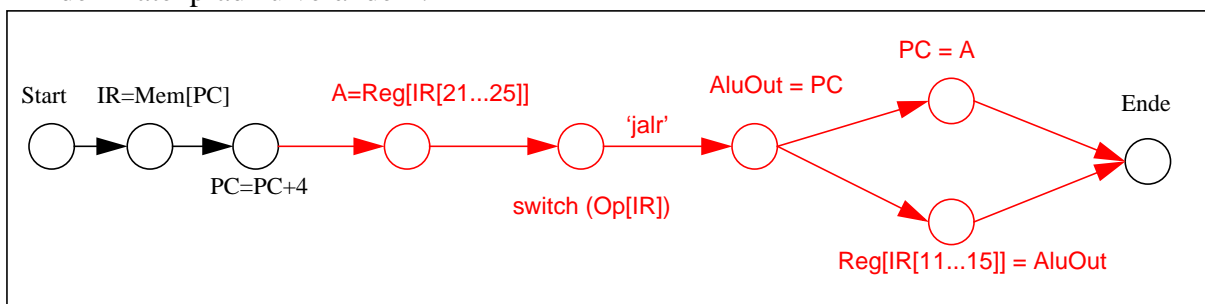
In dieser Aufgabe werden Sie Teile des Kontrollpfades des Ihnen aus der Vorlesung bekannten MIPS-R3000-Prozessors als Mikroprogramm realisieren. Der Datenpfad ist vorgegeben. Für alle Tabellen in den Teilaufgaben gilt, dass Sie ggf. nicht alle Felder ausfüllen müssen. Teile der Aufgaben c) und d) können ohne vorheriges Lösen der Aufgaben a) und b) gelöst werden.



Aufgaben:

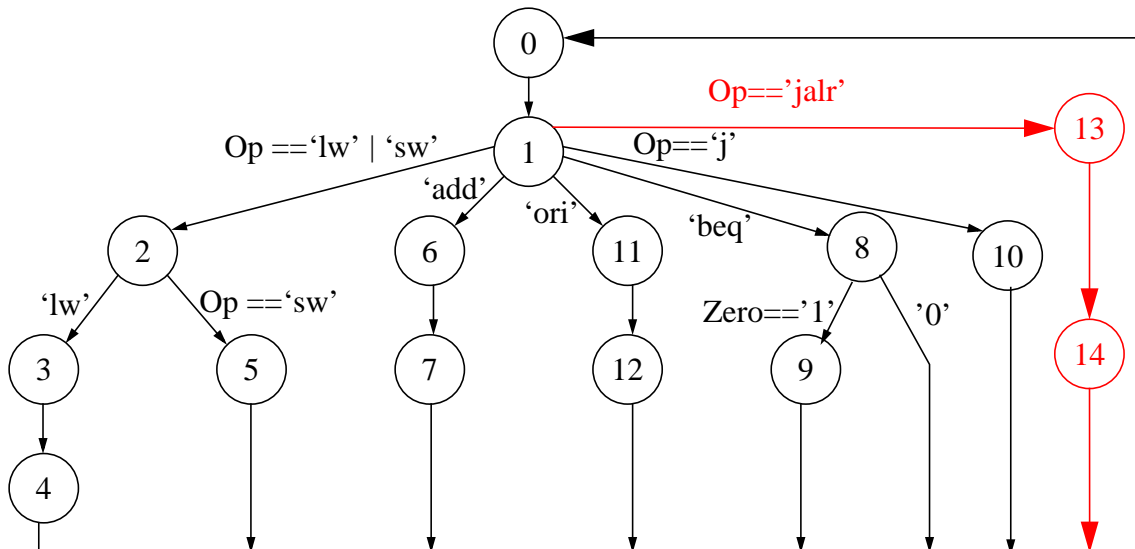
a) (8 Punkte)

Kontrollflussgraph: Der Befehlssatz soll um den `jalr`-Befehl (jump and link register) erweitert werden. Dieser verwendet eine Instruktionkodierung vom Typ R. Der Befehl `jalr` springt zur Adresse, die sich im Register `rs` befindet, und speichert die Adresse der nächsten Instruktion im Register `rd`. Die Felder `rt` und `shamt` sind auf 0 gesetzt. Die ALU kennt die Operation `jalr`, welche den an Eingang `x` angelegten Wert auf den Ausgang der ALU legt. Es gilt die Annahme aus der Vorlesung, dass zwischen Lesen aus dem Speicher, Lesen aus dem Registerfeld und ALU-Operationen die Daten jeweils in Registern zwischengespeichert werden. Ergänzen Sie den Kontrollflussgraphen für `jalr`, ohne den Datenpfad zu verändern.



b) (7 Punkte)

Steuerwerk: Ergänzen Sie den Automaten für das Steuerwerk mit dem jalr-Befehl und fügen Sie die zusätzlich benötigten Zustände in die Tabelle der Ausgangsfunktion ein. Benutzen Sie möglichst wenig neue Zustände. Ist der Zustand eines Steuersignals in einem bestimmten Automatenzustand egal, lassen Sie die entsprechende Tabellenzelle leer.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
ALUSelA	0	0	1				1		1			1		0	1	
ALUSelB	1	3	3				0		0			2				
ExtOp		sign	sign									zero				
ALUOp	add	add	add				add		sub			or		jalr	jalr	
TarWr		1							0							
ALUOutWr			1				1					1		1		
lorD	0			1												
MemWr	0	0	0	0	0	1	0	0	0	0	0	0	0	0		
IRWr	1	0	0	0			0					0		0		
MemOutWr				1												
RegDst					0			1					0		1	
MemtoReg					1			0					0		0	
RegWr	0	0	0	0	1	0	0	1	0	0	0	0	1	0	1	
RegOutWr		1	0													
PCSrc	0									1	2				0	
PCWr	1	0	0	0	0	0	0	0	0	1	1	0	0	0	1	

c) (5 Punkte)

Mikroprogramme: Eine binäre Kodierung der symbolischen Ausgänge ist in den folgenden Tabellen vorgegeben. Ergänzen Sie die Tabellen in Abbildung 1 mit den Sequenzen der Mikroinstruktionen, die zur Abarbeitung aller Phasen der Befehle `ori` und `jalr` nötig sind (Holen der Instruktion / Dekodierung / Ausführung / Speichern). Setzen Sie dabei Signale, deren Wert egal ist, auf 0. Die Zeilen 0 und 1 sind jeweils schon eingetragen. (Hinweis: der Tabelleneintrag "Referenznummer" wird erst in Teilaufgabe d) benötigt.)

	ALUOp ₁	ALUOp ₀
jalr	0	0
add	0	1
sub	1	0
or	1	1

	ALUSelB ₁	ALUSelB ₀
0	0	0
1	0	1
2	1	0
3	1	1

	PCSrc ₁	PCSrc ₀
0	0	0
1	0	1
2	1	0

	EXTOp ₀
sign	0
zero	1

Mikroprogramm (binär) für den Befehl `ori`:

Zeile	Referenznummer	ALU Control								Memory Control				Register Control				PC Control			
		ALUSelA	ALUSelB ₁	ALUSelB ₀	ExtOp	ALUOp ₁	ALUOp ₀	TarWr	ALUOutWr	lorD	MemWr	IRWr	MemOutWr	RegDst	MemtoReg	RegWr	RegOutWr	PCSrc ₁	PCSrc ₀	PCWr	
0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1
1	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1
2	2	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
4	4																				
5	5																				

Mikroprogramm (binär) für den Befehl `jalr`:

Zeile	Referenznummer	ALU Control								Memory Control				Register Control				PC Control			
		ALUSelA	ALUSelB ₁	ALUSelB ₀	ExtOp	ALUOp ₁	ALUOp ₀	TarWr	ALUOutWr	lorD	MemWr	IRWr	MemOutWr	RegDst	MemtoReg	RegWr	RegOutWr	PCSrc ₁	PCSrc ₀	PCWr	
0	6	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1
1	7	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1
2	8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
3	9	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1
4	10																				
5	11																				

Abbildung 1

d) (5 Punkte)

Mikroprogramm- und Sprungspeicher: Die Befehle `ori` und `jalr`, die Sie bisher getrennt in Teilaufgabe c) behandelt haben, sollen nun von der mikroprogrammgesteuerten Architektur in Abbildung 2 verarbeitet werden können. Überlegen Sie sich, welche Mikroinstruktionen Sie aus Teilaufgabe c) benötigen und wie Sie diese Zeilen in dem Mikroprogramm Speicher in Abbildung 2 anordnen würden, um mit Hilfe des Sprungspeichers A und des zusätzlichen Signals `MPC-Source` mit möglichst wenig Programmzeilen auszukommen. Tragen Sie dazu in Abbildung 2 die Referenznummern aus Abbildung 1 in die Mikroprogramm Speichertabelle ein und ergänzen Sie das `MPC-Source`-Feld sowie die Einträge im Sprungspeicher A. **Graue Felder nicht ausfüllen!**

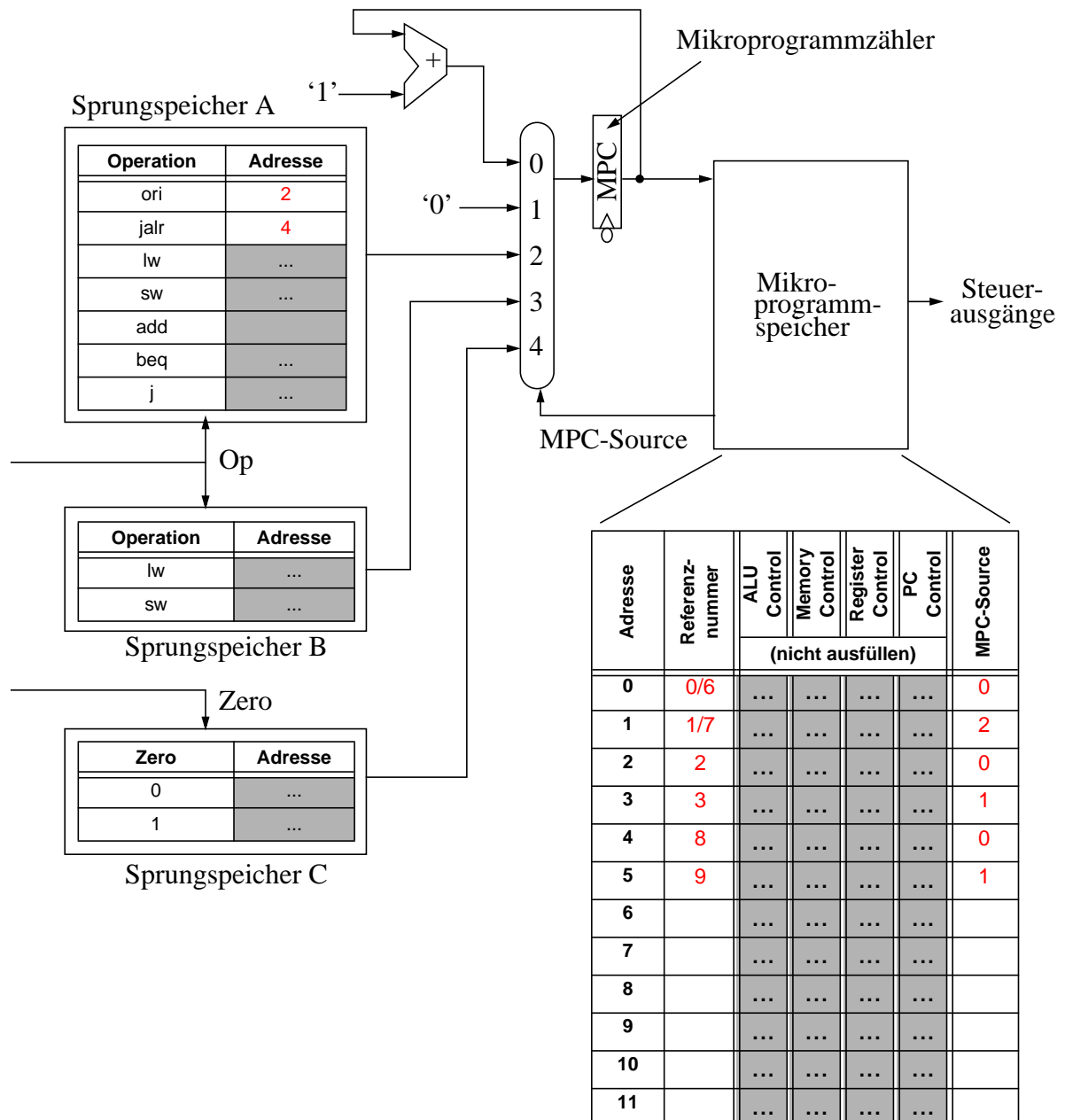


Abbildung 2