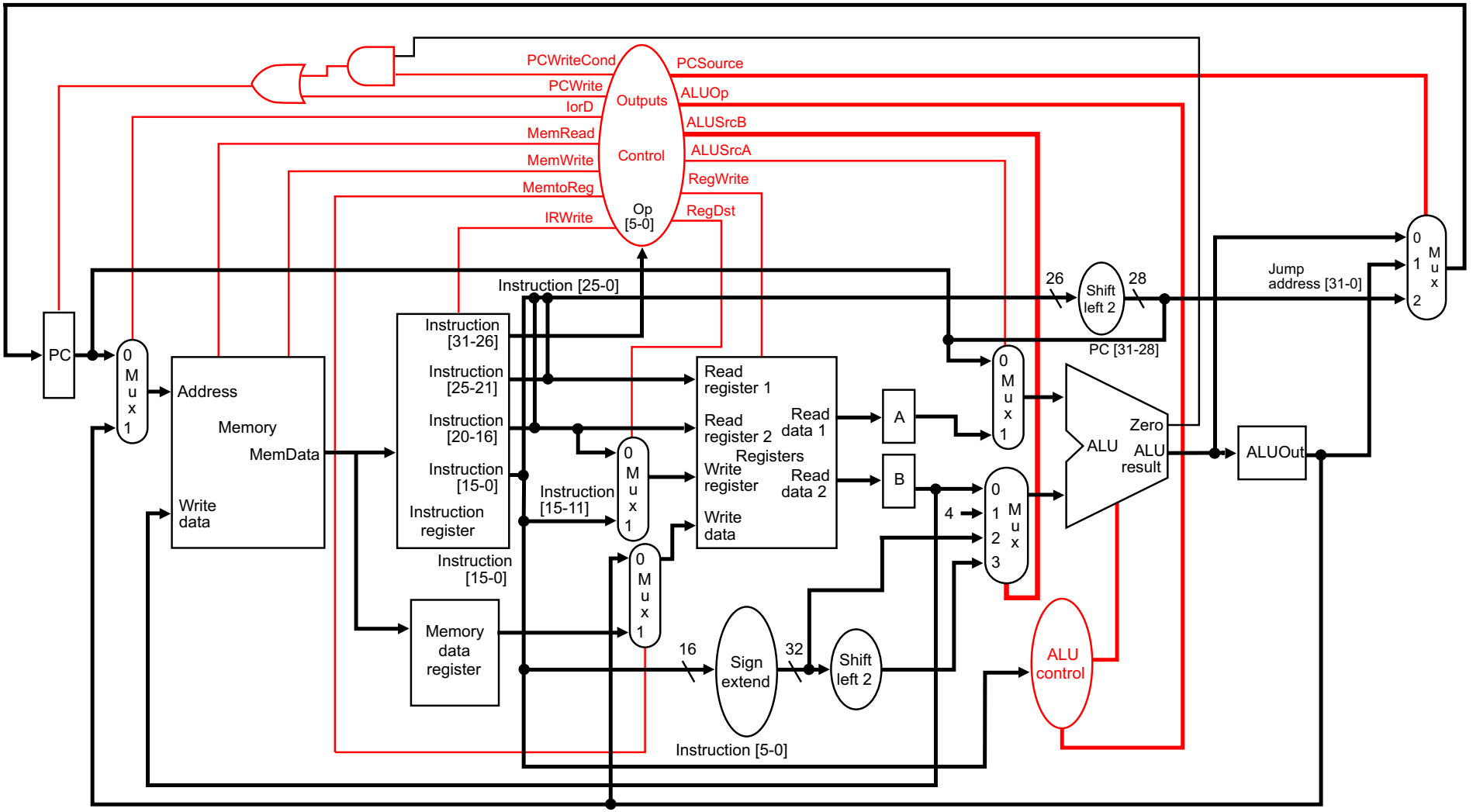


Architektura typu *multi–cycle*

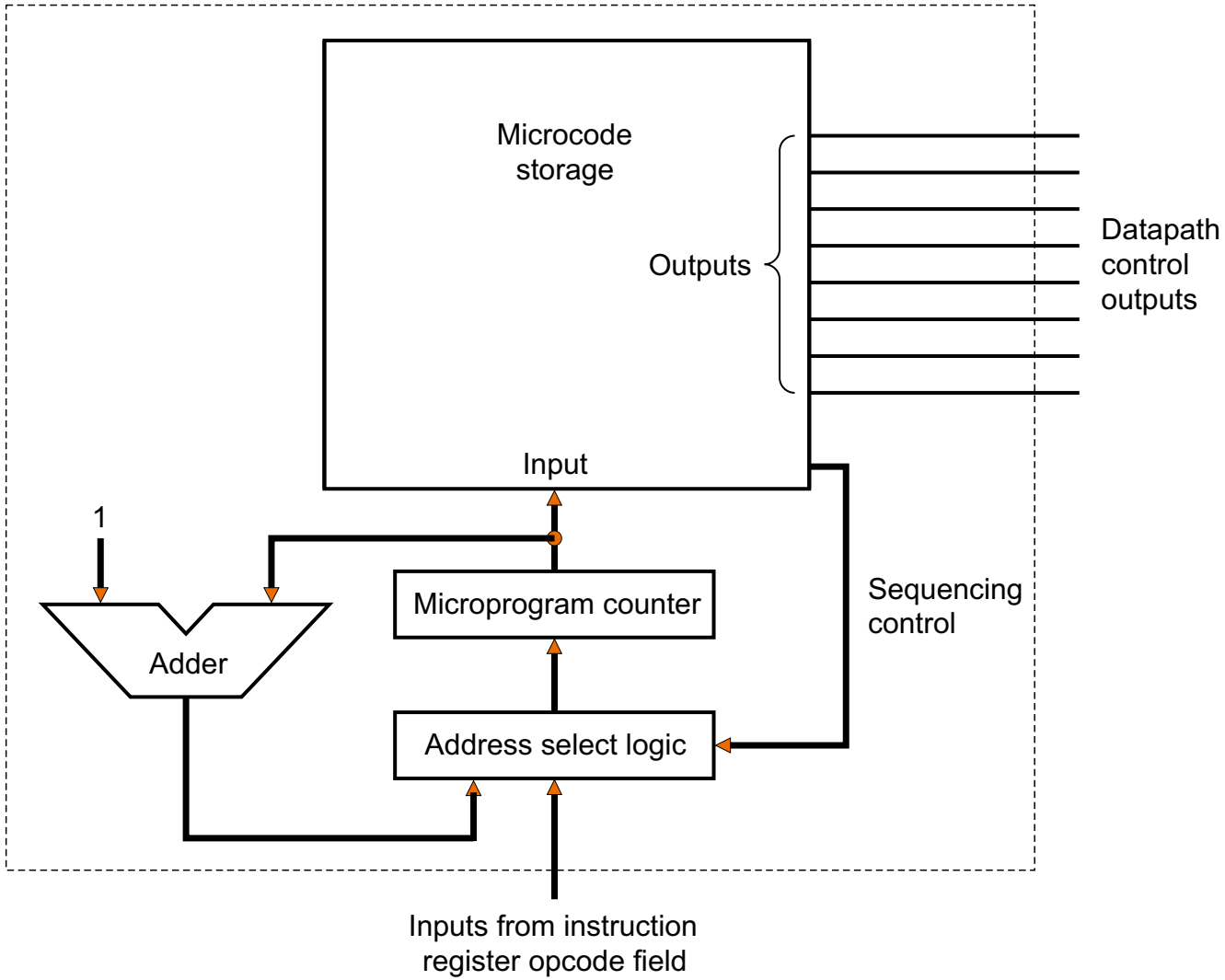
- Idea sterowania mikroprogramowalnego opiera się na realizacji sterowania w postaci pamięci o n-bitowych słowach zwanych mikroinstrukcjami, które reprezentują wartości linii sterujących mikroprocesora, i która jest adresowana przez licznik mikroinstrukcji.

- W kolejnych cyklach zegara, mikroinstrukcje są podawane na wyjście pamięci zapewniając prawidłową realizację każdej instrukcji.

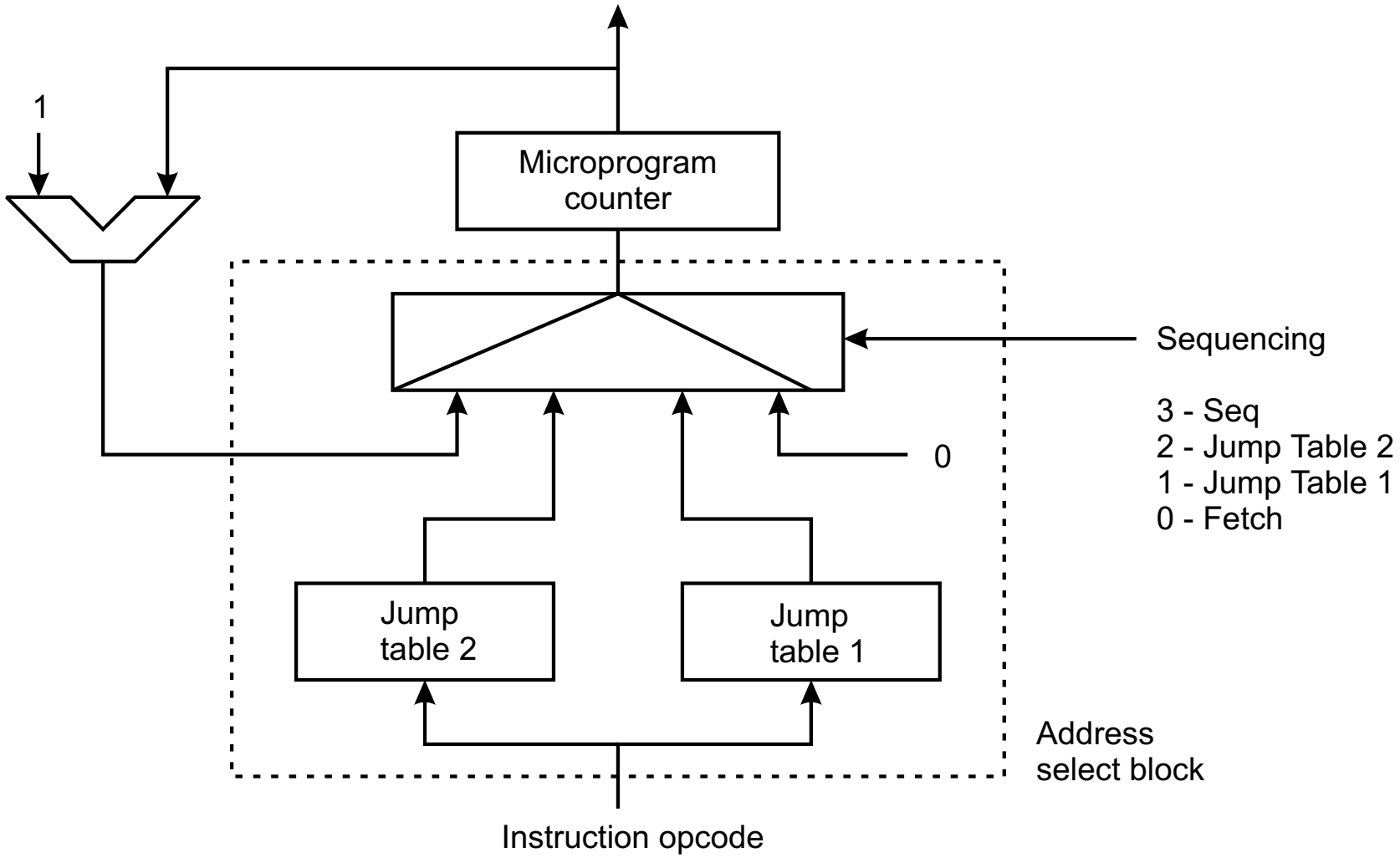
- Sekwencje mikroinstrukcji realizujące wszystkie instrukcje procesora nazywa się mikroprogramem.

- Niektóre mikroinstrukcje mogą być wspólne dla instrukcji mikroprocesora, co ogranicza pamięć mikroprogramu, a liczba wszystkich mikroinstrukcji może być równa liczbie stanów maszyny stanowej.

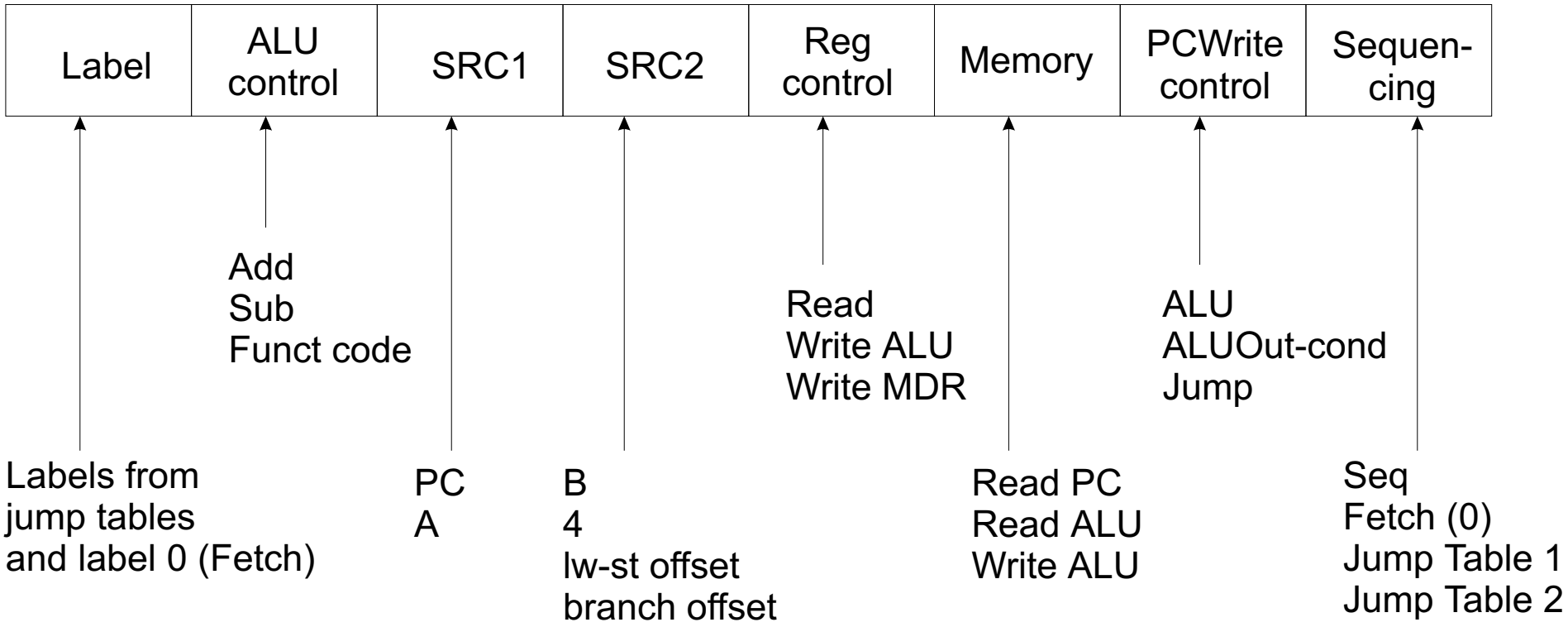
- Mikroprogram musi mieć możliwość zarówno wykonywania sekwencyjnego, jak i wykonywania skoków bezwarunkowych i warunkowych wewnątrz mikrokodu, w zależności od typu (opcode'u) wykonywanej instrukcji.



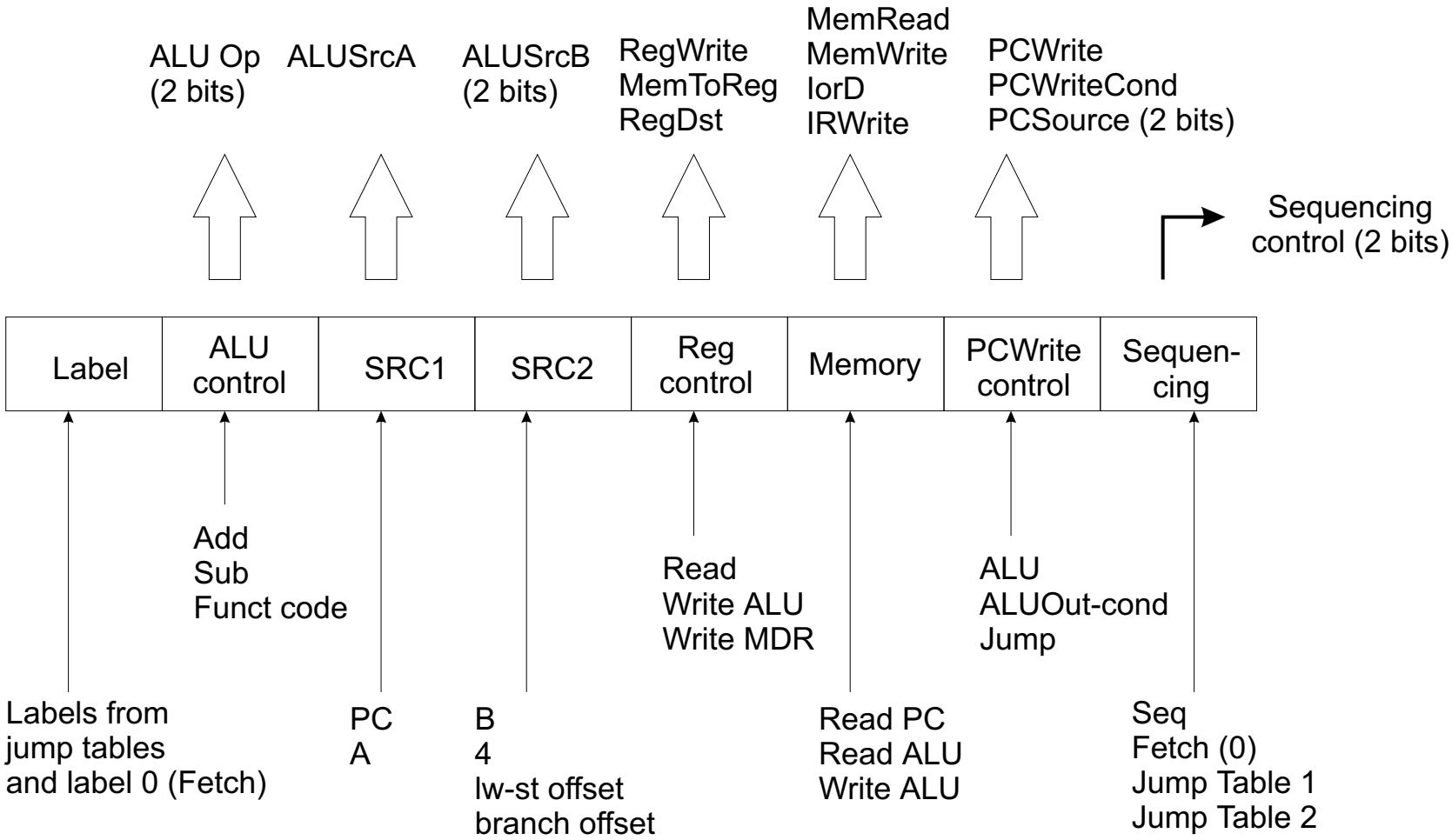
Mikroprogramowalna jednostka sterująca



Sterowanie mikroprogramu — *Sequencing*



Linia mikroprogramu



Linie sterujące pól mikrokodu

Label	ALU control	SRC1	SRC2	Reg control	Memory	PCWrite control	Sequen-cing
Fetch	Add	PC	4		Read PC	ALU	Seq
	Add	PC	branch offset	Read			Jump Table1

Mikroprogram dla fazy *Fetch* i *Decode*

Label	ALU control	SRC1	SRC2	Reg control	Memory	PCWrite control	Sequen-cing
Fetch	Add	PC	4		Read PC	ALU	Seq
	Add	PC	branch offset	Read			Jump Table1
Mem1	Add	A	ls-sw offset				Jump Table2
LW2					ReadALU		Seq
				WriteMDR			Fetch
SW2					WriteALU		Fetch
Rtype1	Func cod	A	B				Seq
				WriteALU			Fetch
Beq1	Sub	A	B			ALUOut cond.	Fetch
Jump1						Jump	Fetch

Kompletny mikroprogram

Jump Table 1

Opcode	Label
Rtype	Rtype1
Jump	Jump1
Beq	Beq1
Load	Mem1
Store	Mem1

Jump Table 2

Opcode	Label
Load	LW2
Store	SW2

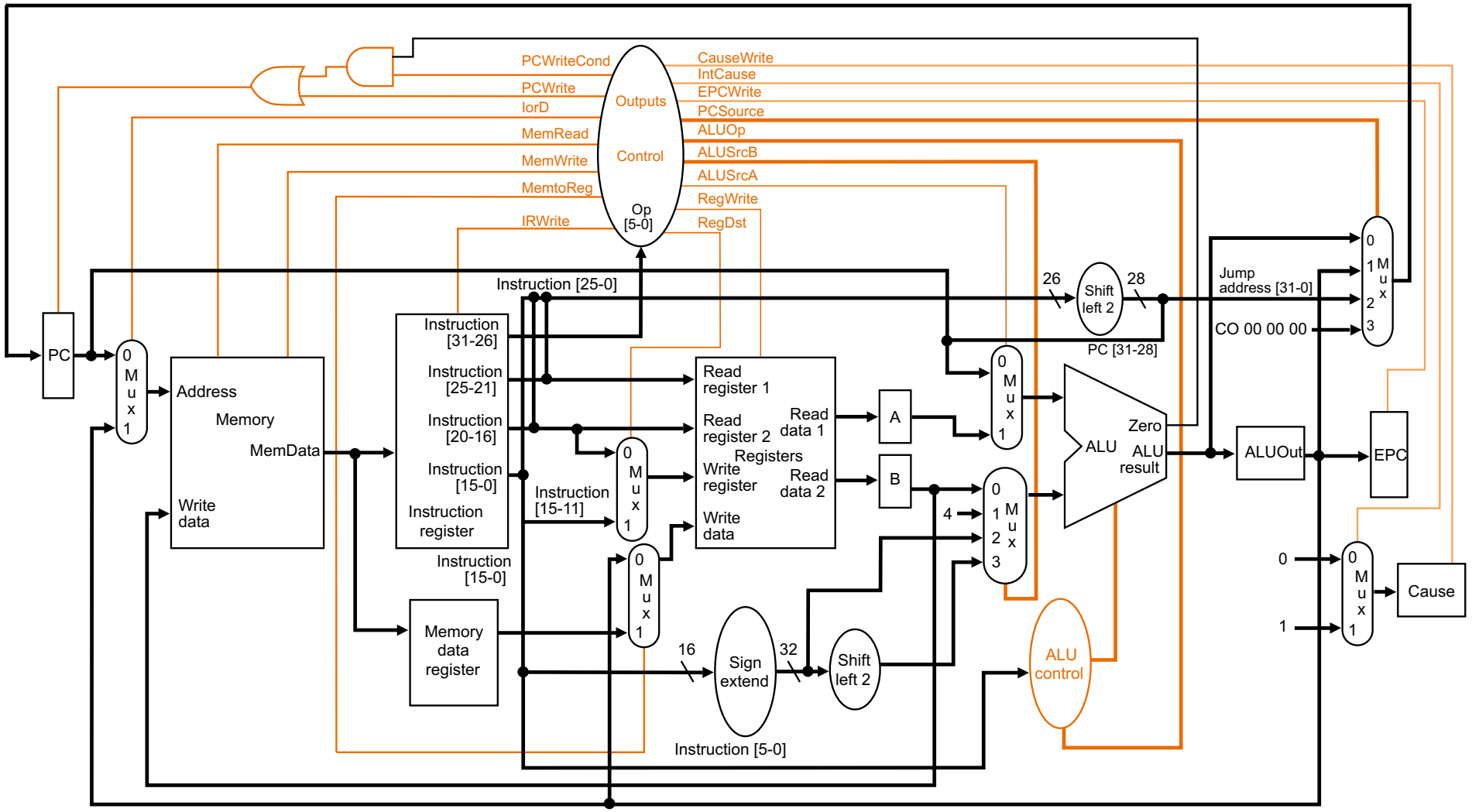
Tablice skoków mikroprogramu

Sytuacje wyjątkowe — *Exceptions/Interrupts*

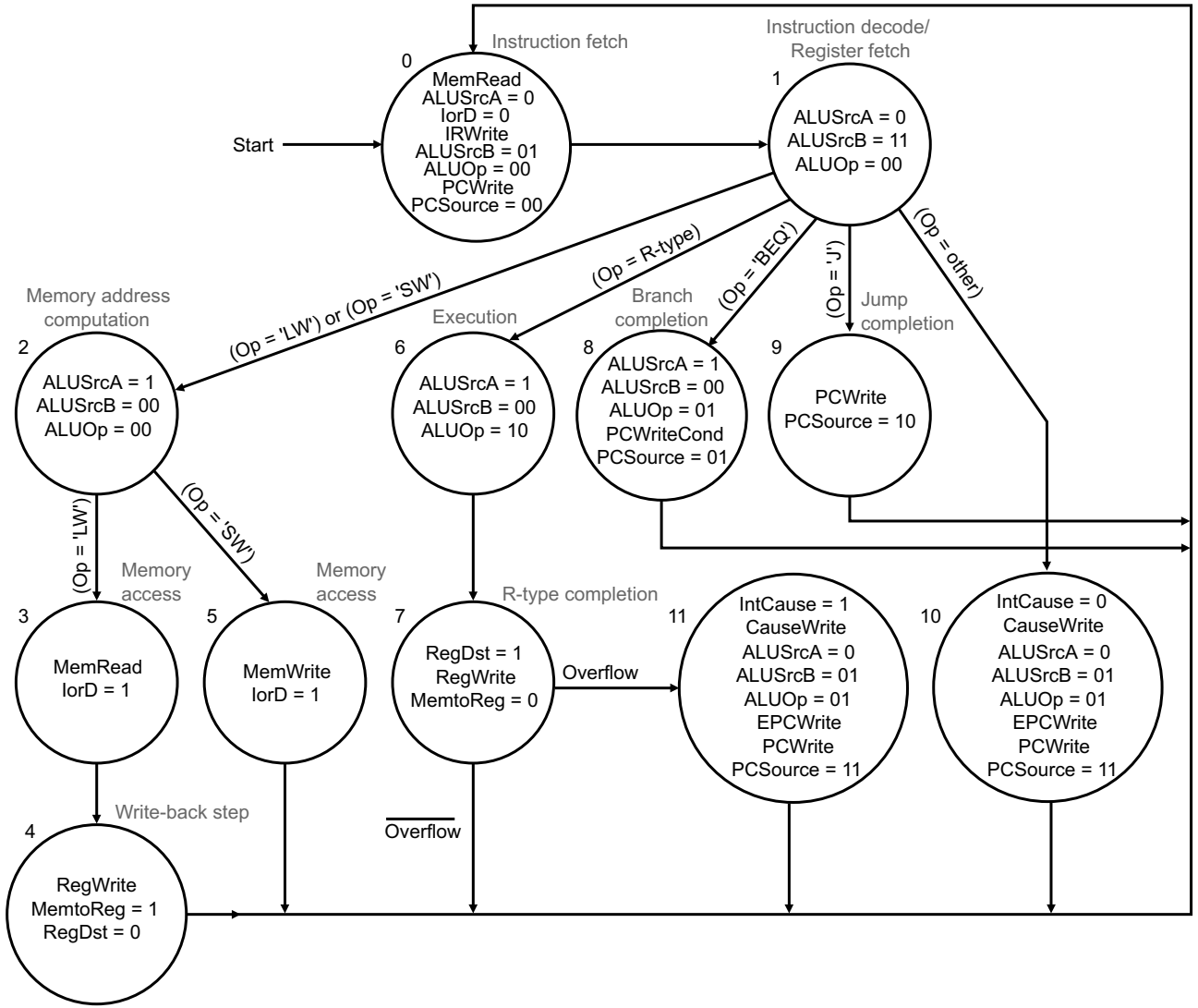
- Sytuacja wyjątkowa — nieoczekiwane zdarzenie, wymagające zmiany sekwencji wykonywanych instrukcji, tj. wywołania procedury obsługi zdarzenia
- Dwa rozwiązania obsługi sytuacji wyjątkowych:
  - skok do adresu związanego z daną sytuacją wyjątkową
  - zapisanie do rejestru numeru sytuacji wyjątkowej i skok do wspólnego adresu procedury obsługi
- Realizacja sytuacji wyjątkowych (wariant 2)
  - rejestr EPC (*Exception Program Counter*)
  - rejestr Cause — numer sytuacji wyjątkowej
  - linie sterujące *CauseWrite, IntCause, EPCWrite*

Realizacja sytuacji wyjątkowych

- nielegalna instrukcja — Cause=0 — próba wykonania instrukcji, której pole *opcode* nie odpowiada żadnej instrukcji
- przepełnienie arytmetyczne — Cause=0 — generacja bitu *overflow* przez sumator w czasie wykonywania instrukcji arytmetycznych
- adres wspólnej procedury obsługi $C0000000 do PC



Implementacja sytuacji wyjątkowych w architekturze *multi-cycle*



Sterowanie z uwzględnieniem sytuacji wyjątkowych