



# Zastosowania mikrokontrolerów w przemyśle

---

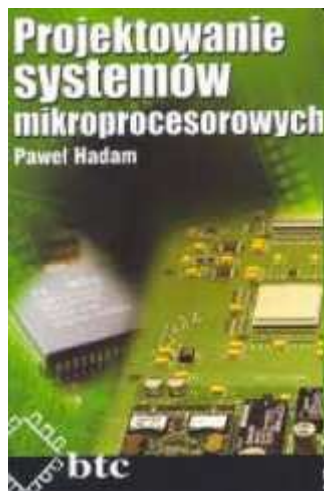
Cezary MAJ

Katedra Mikroelektroniki i Technik Informatycznych

# Literatura

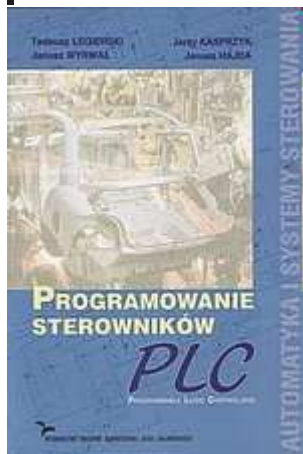


- Ryszard Pełka: *Mikrokontrolery - architektura, programowanie, zastosowania*

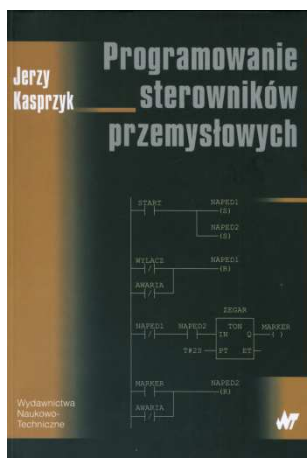


- *Projektowanie systemów mikroprocesorowych* - Paweł Hadam

# Literatura



- *Programowanie sterowników PLC* - Tadeusz Legierski, Janusz Wyrwał, Jerzy Kasprzyk, Janusz Hajda



- *Programowanie sterowników przemysłowych* - Jerzy Kasprzyk.



# Mikrokontroler

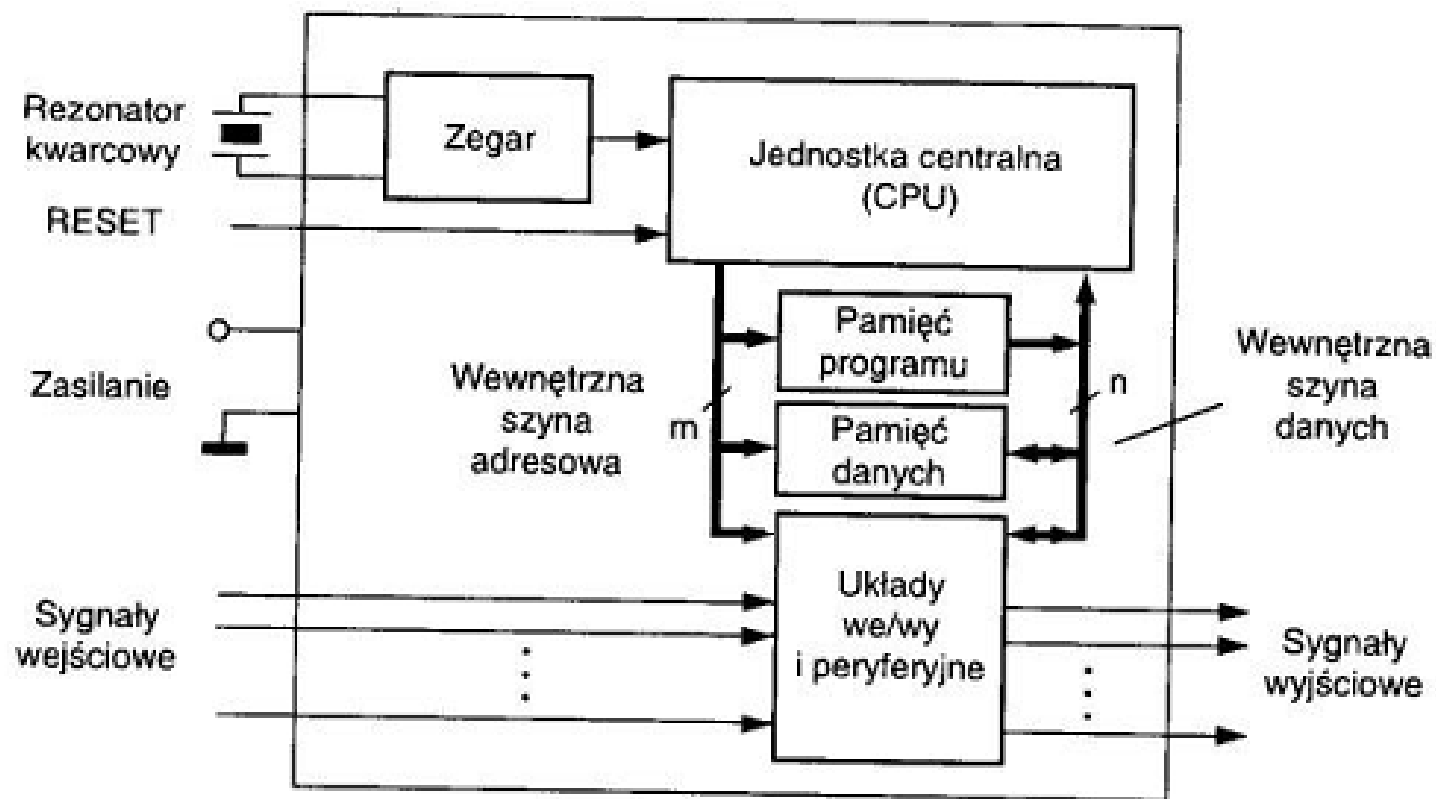
---

Układ scalony z mikroprocesorem

- Autonomiczna praca
- Rozbudowany system komunikacji z otoczeniem

Mikroprocesor – do pracy wymaga elementów zewnętrznych

# Mikrokontroler



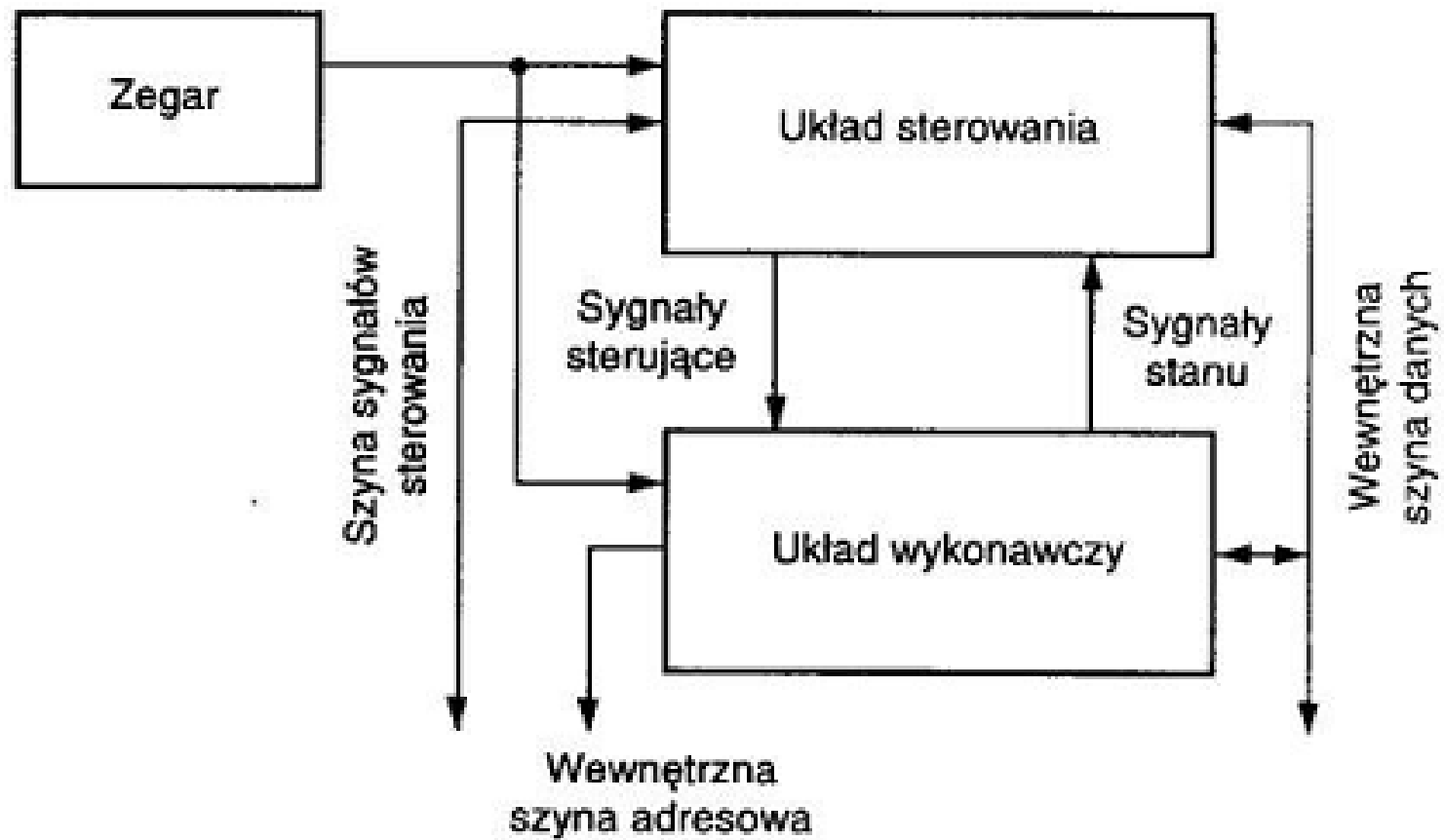


# Mikrokontroler

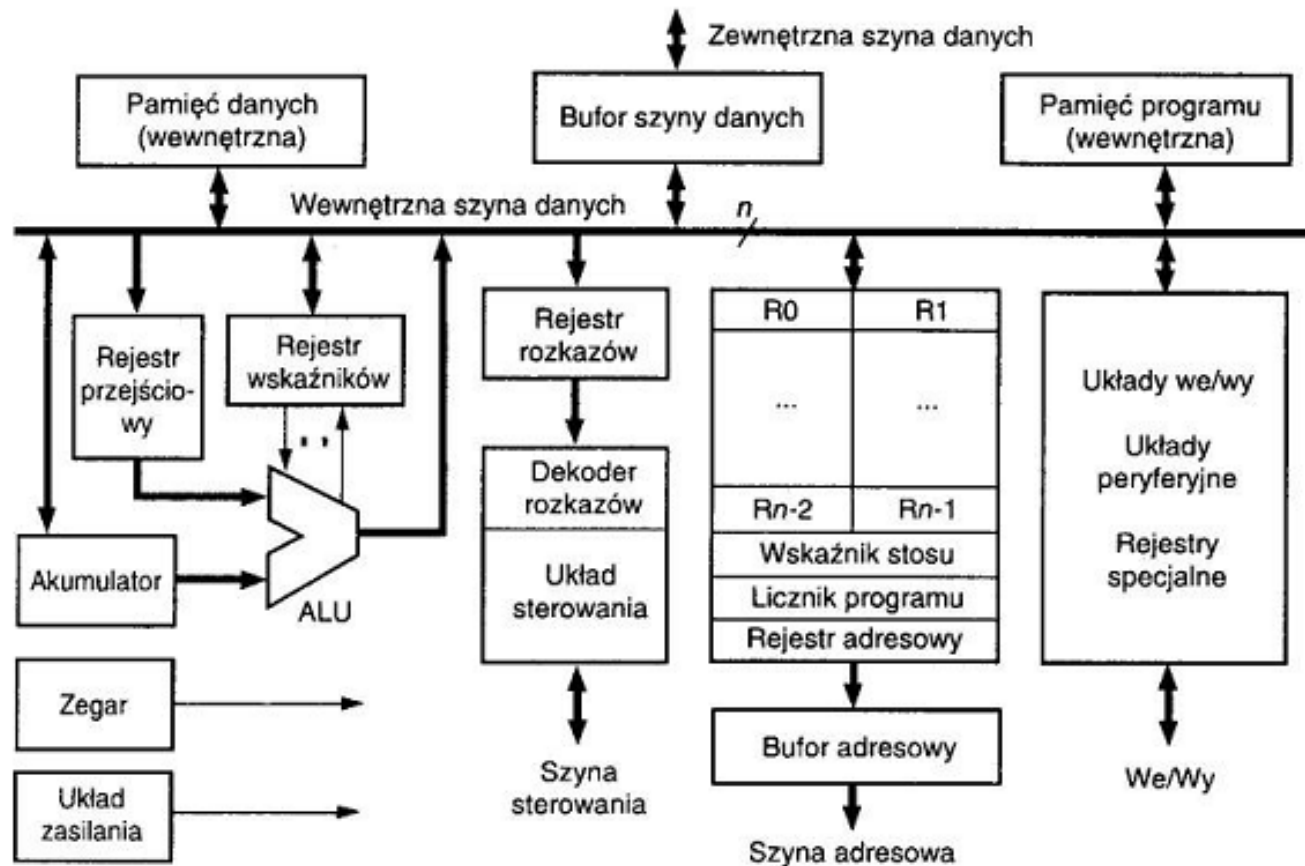
---

- $n$  – szyna danych, określa bitowość mikrokontrolera (ale nie zawsze!)
- $m$  – szyna adresowa, określa maksymalny rozmiar pamięci do której mikrokontroler ma dostęp (przestrzeń adresowa)
- Kierunek szyny

# Architektura

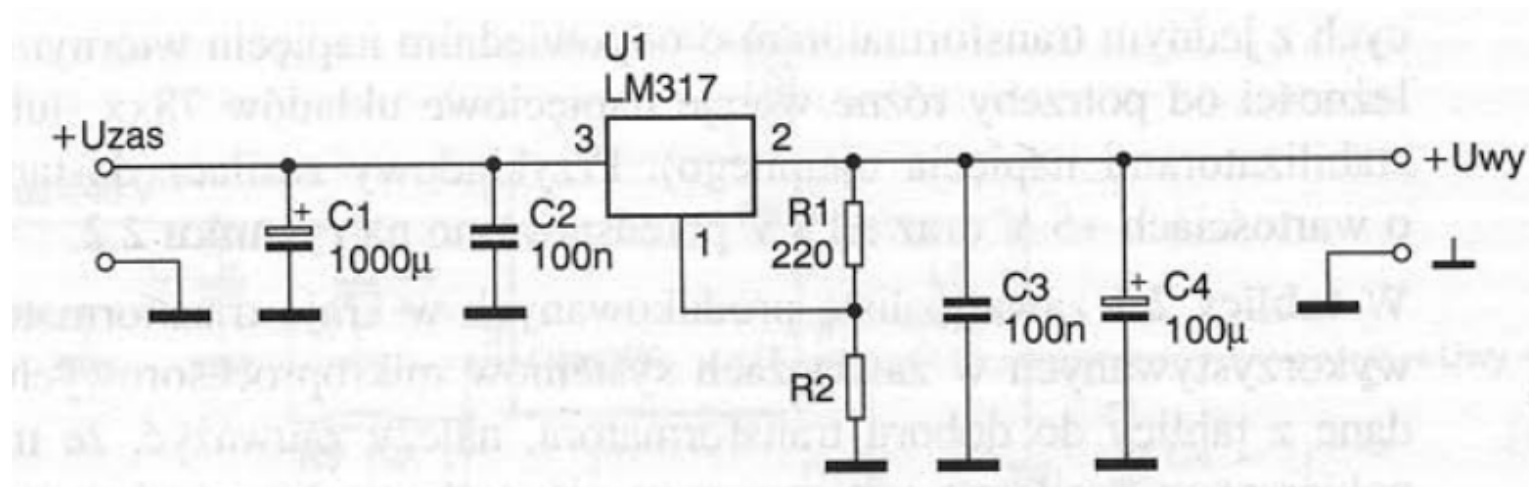
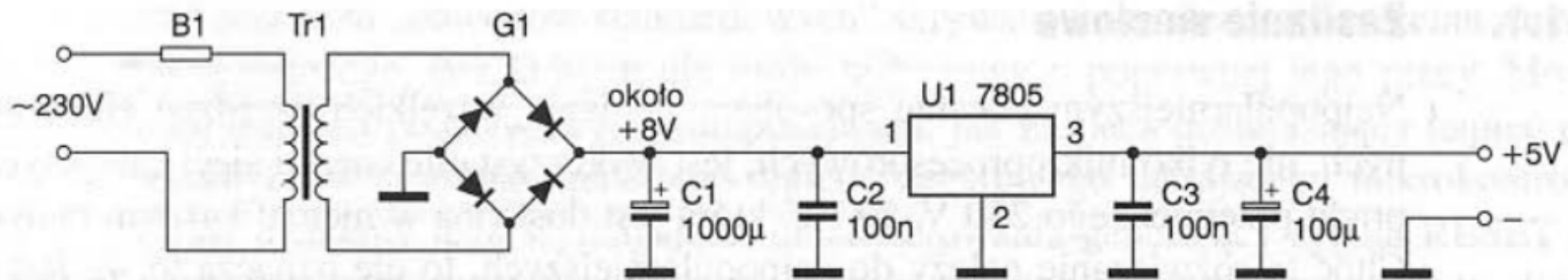


# Architektura

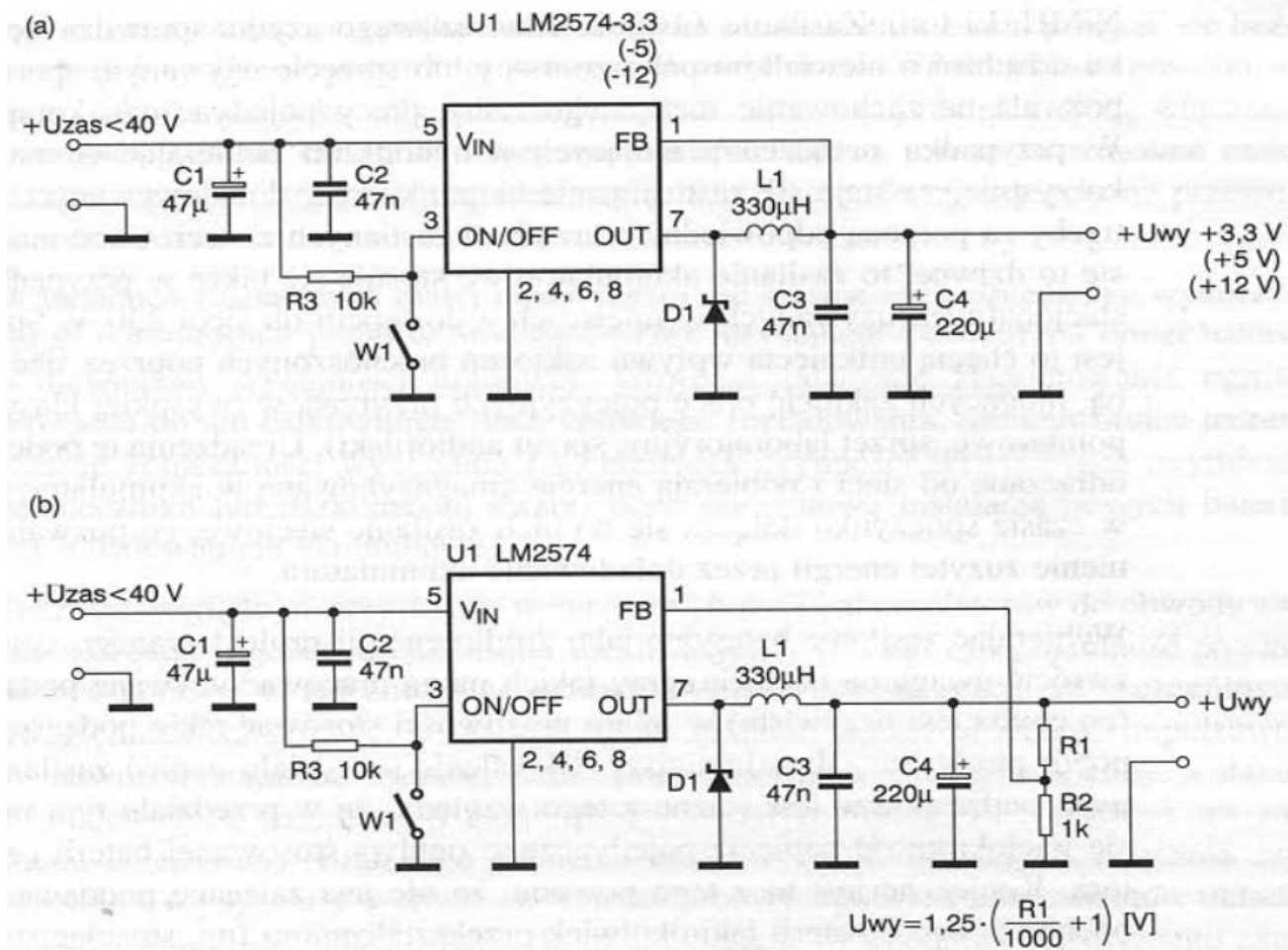




# Układ zasilania - liniowy



# Układ zasilania - impulsowy





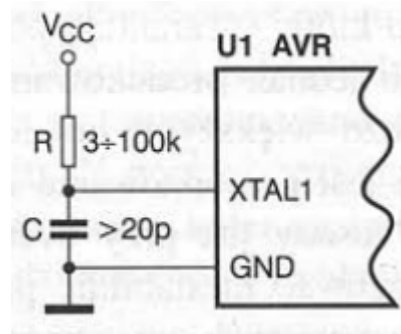
# Układ zasilania - ogniwo

---

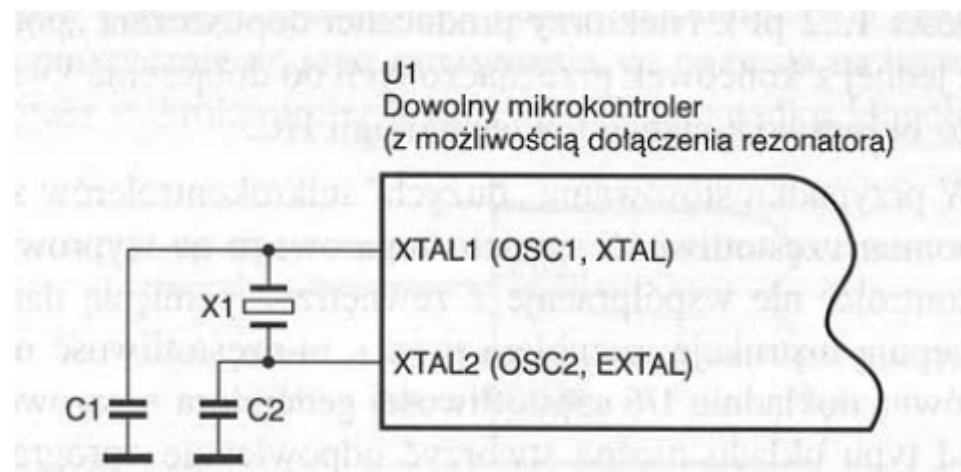
- Bateria, akumulator (Ni-MH, Li-Ion, Li-Po)
- Przetwornica podwyższająca, obniżająca
- Zasilanie sieciowo-akumulatorowe

# Generator sygnału CLK

- Wbudowane
  - RC, mało stabilne, niska częstotliwość



- Zewnętrzne
- Pętla PLL





# Układ wykonawczy

---

- ALU
- Akumulator
- Rejestr tymczasowy
- Rejestr wskaźników (CZNV)



# Układ sterowania

---

- Sterowanie układem wykonawczym
- Rejestr rozkazów
- Dekoder rozkazów
- Licznik programu
- Rejestry specjalne (SFR), wskaźnik stosu (SP)
- Kontroler szyn, przerwań, ...

# Rejestry

- Pamięć „podręczna” do zapisu danych
- Szybszy zapis niż zapis do pamięci danych

R15
R14
R13
R12
R11
R10
R09
R08

16-bitowe  
rejestry globalne

R7H	R7L
R6H	R6L
R5H	R5L
R4H	R4L

8- i 16-bitowe  
rejestry globalne

R3H	R3L
R2H	R2L
R1H	R1L
R0H	R0L

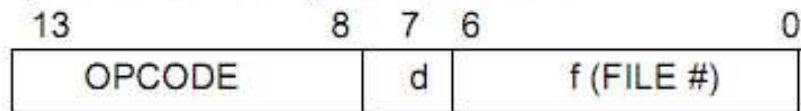
4 zestawy (banki)  
rejestrów 8- i 16-bitowych



# Operacje

---

## Byte-oriented file register operations

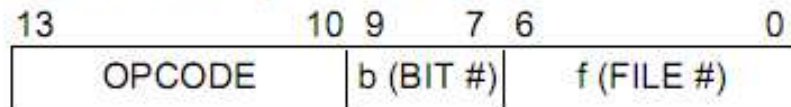


d = 0 for destination W

d = 1 for destination f

f = 7-bit file register address

## Bit-oriented file register operations

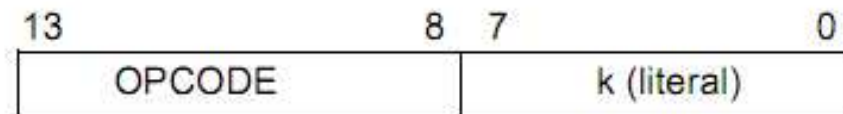


b = 3-bit bit address

f = 7-bit file register address

## Literal and control operations

### General



k = 8-bit immediate value

### CALL and GOTO instructions only

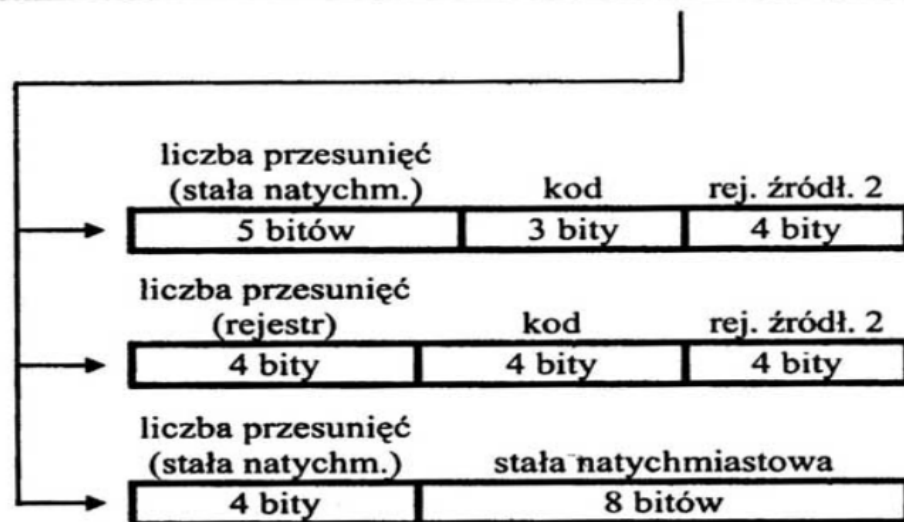


k = 11-bit immediate value



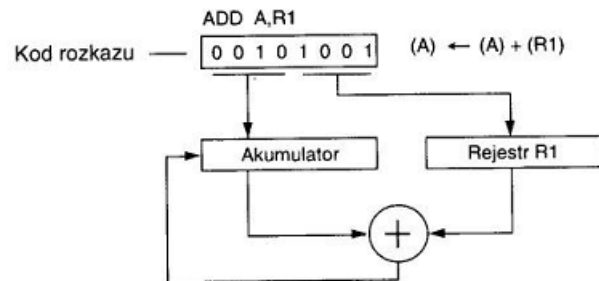
# Operacje ARM

warunek	kod grupy r.	kod rozkazu	uaktualnienie CPSR	rejestr docelowy	pierwszy operand	pole Operand_2
4 bity	3 bity	4 bit	1bit	4 bity	4 bity	12 bitów

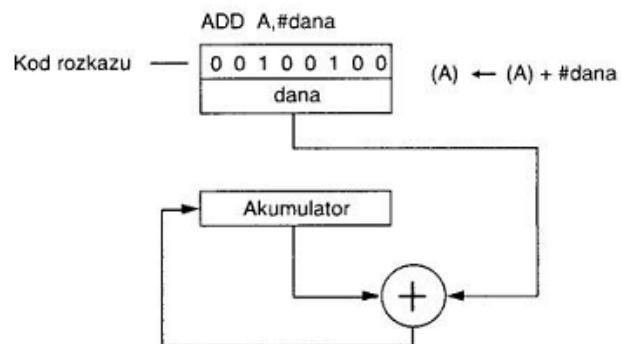


# Adresowanie

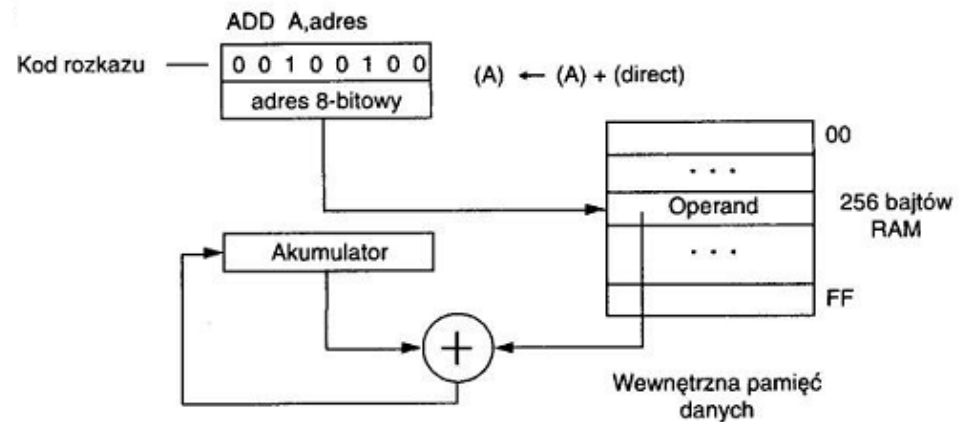
## ■ Rejestrowe



## ■ Natychmiastowe

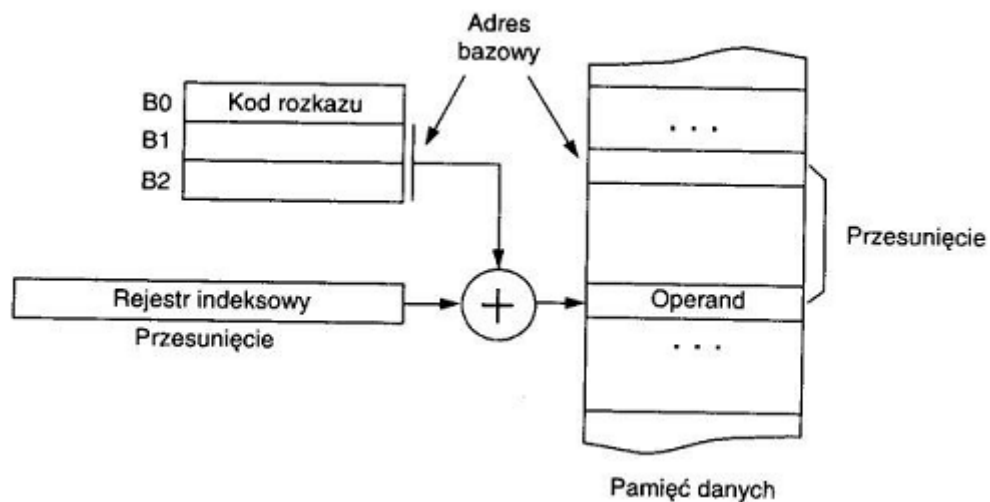


## ■ Bezpośrednie

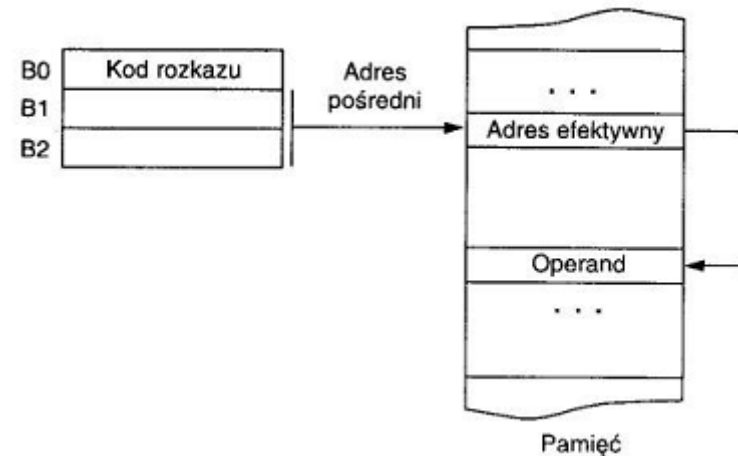


# Adresowanie

## ■ Indeksowe



## ■ Pośrednie





# Adresowanie ARM

---

- ◆ Architektura ARM udostępnia 11 podstawowych trybów adresowania:
  - ◆ Adresowanie natychmiastowe (ang. immediate) #<immediate>, np. #13,
  - ◆ Adresowanie rejestrowe bezpośrednio (ang. register) <Rm>, np. r7,
  - ◆ Adresowanie rejestrowe z przesunięciem (ang. register with offset) <Rm>, rot #<shift\_imm>, np. r0, LSL #4,
  - ◆ Adresowanie rejestrowe pośrednie (ang. register indirect),
  - ◆ Adresowanie rejestrowe pośrednie z indeksowaniem (ang. register indirect pre-indexed with no write-back),
  - ◆ Adresowanie rejestrowe pośrednie z preindeksowaniem (ang. register indirect pre-indexed with write-back),
  - ◆ Adresowanie rejestrowe pośrednie z postindeksowaniem (ang. register indirect post-indexed with write-back),
  - ◆ Adresowanie względem licznika programu (ang. Program Counter register indirect).
    - ◆ Pośrednie względem PC,
    - ◆ Pośrednie względem PC z indeksowaniem,
    - ◆ Pośrednie względem PC z preindeksowaniem,
    - ◆ Pośrednie względem PC z postindeksowaniem.



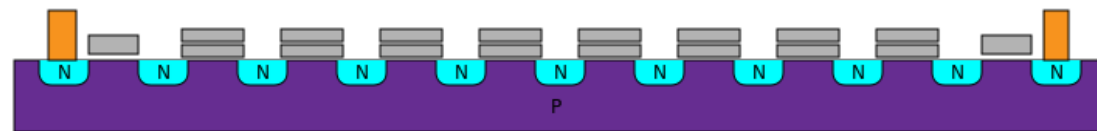
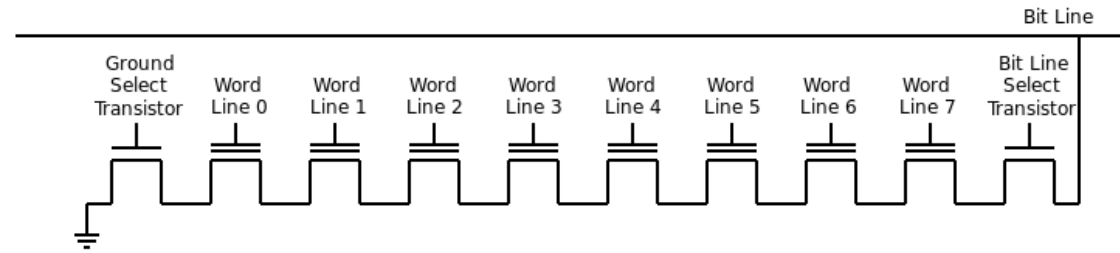
# Pamięć

---

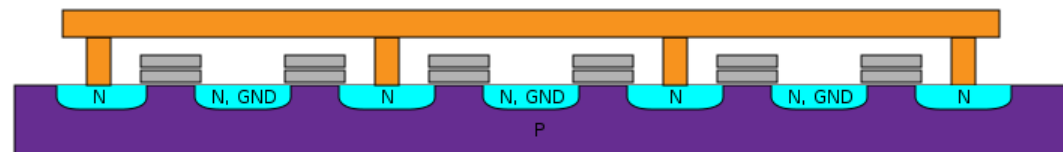
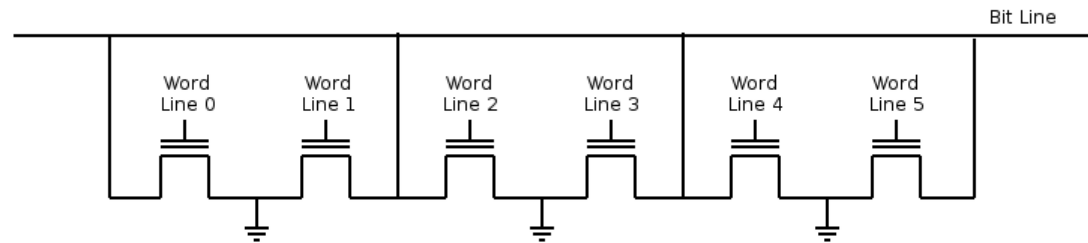
- RAM
- ROM
- EEPROM
  - Flash –operacje na wielu komórkach za pomocą jednej operacji
    - NOR (dostęp do każdej komórki, wolny transfer - firmware)
    - NAND (szybki transfer blokowy – pamięć przenośna)

# Pamięć

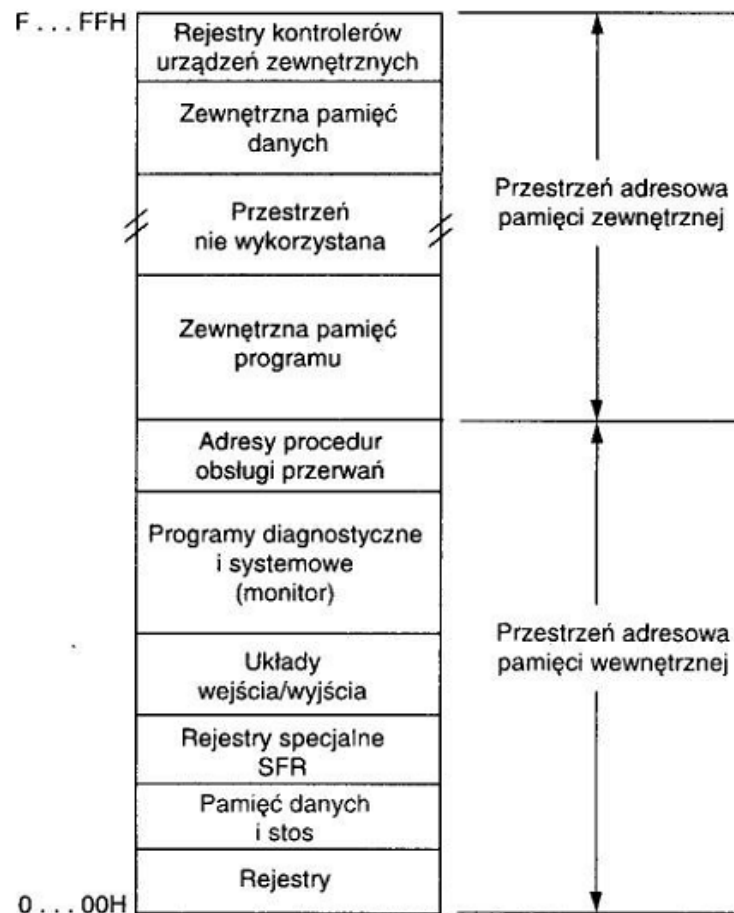
NAND



NOR

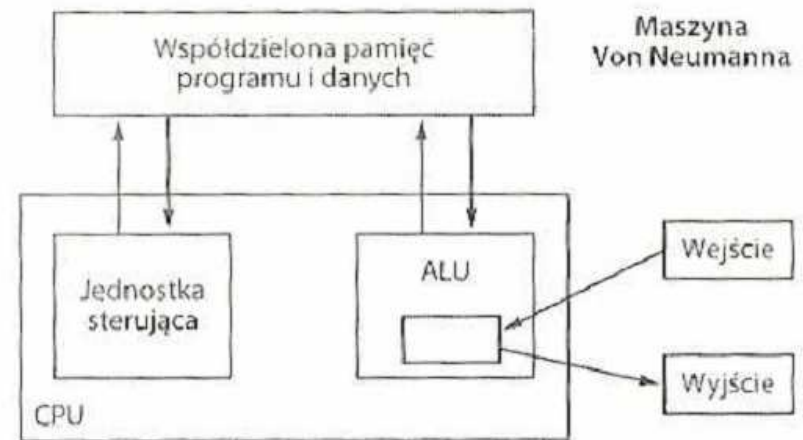


# Mapa pamięci

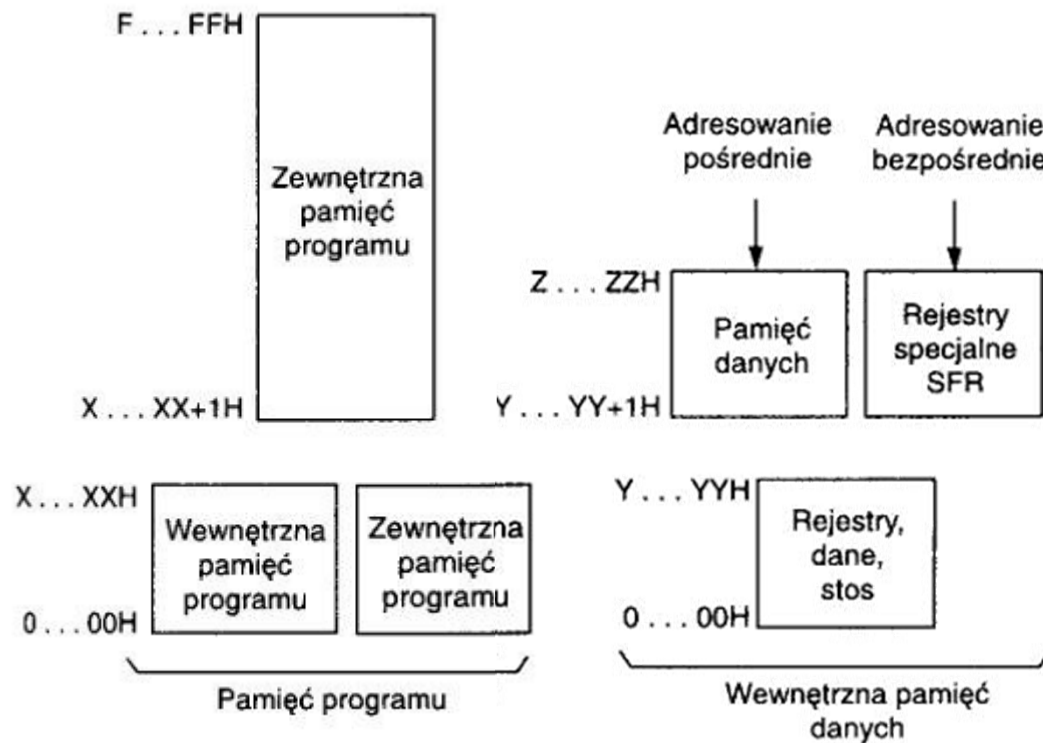


## Architektura Von-Neumanna

Wspólna szyna dla programu i danych



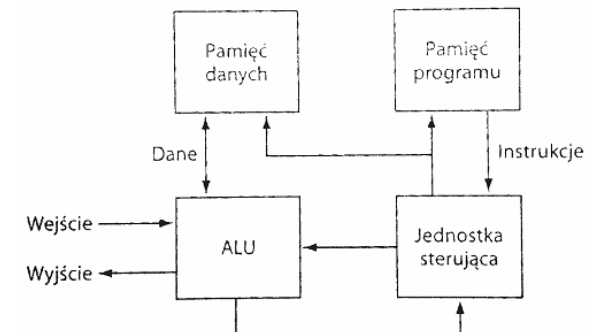
# Mapa pamięci



## Architektura Harwardzka

Rozdzielona szyna dla programu i danych

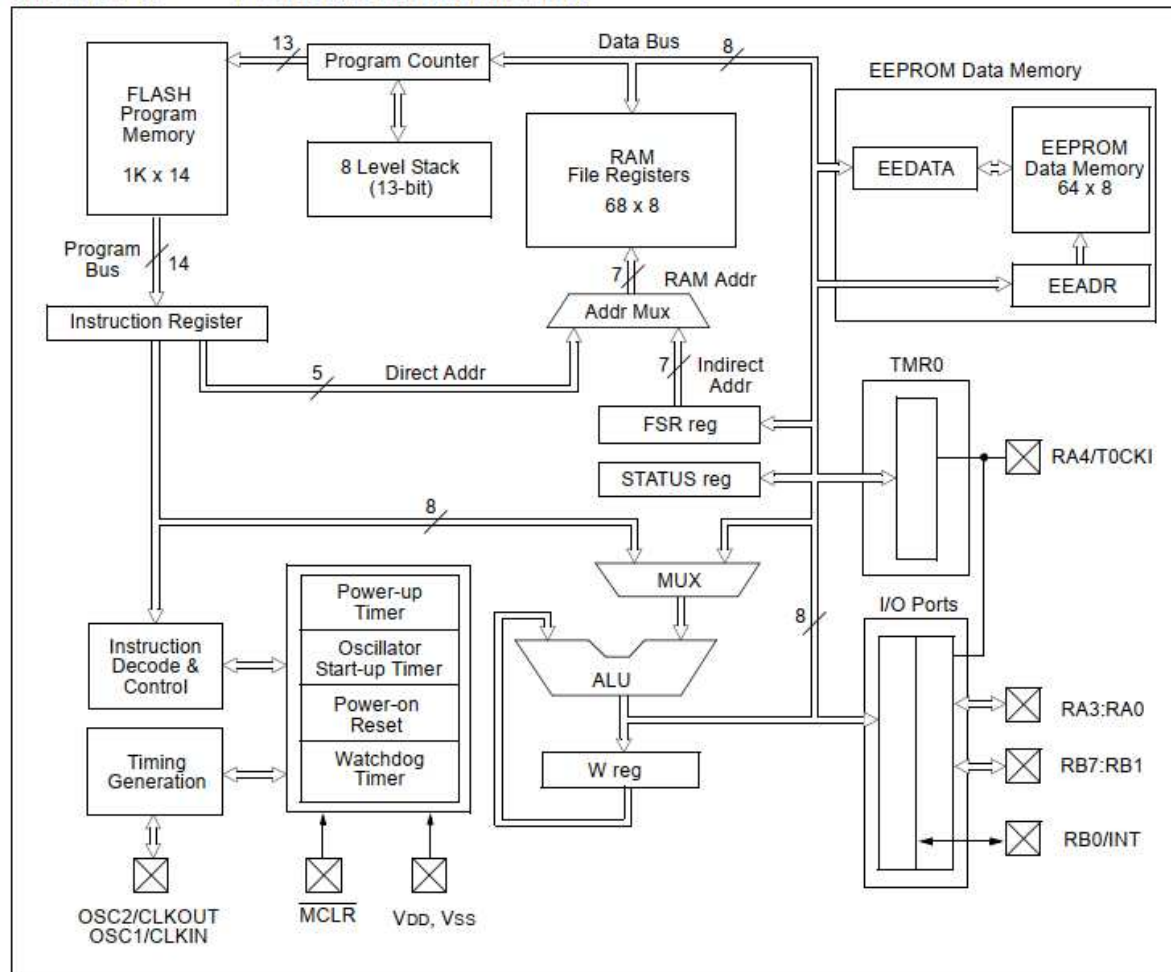
Możliwość jednoczesnego wykonywania i pobierania rozkazu (przetwarzanie potokowe)



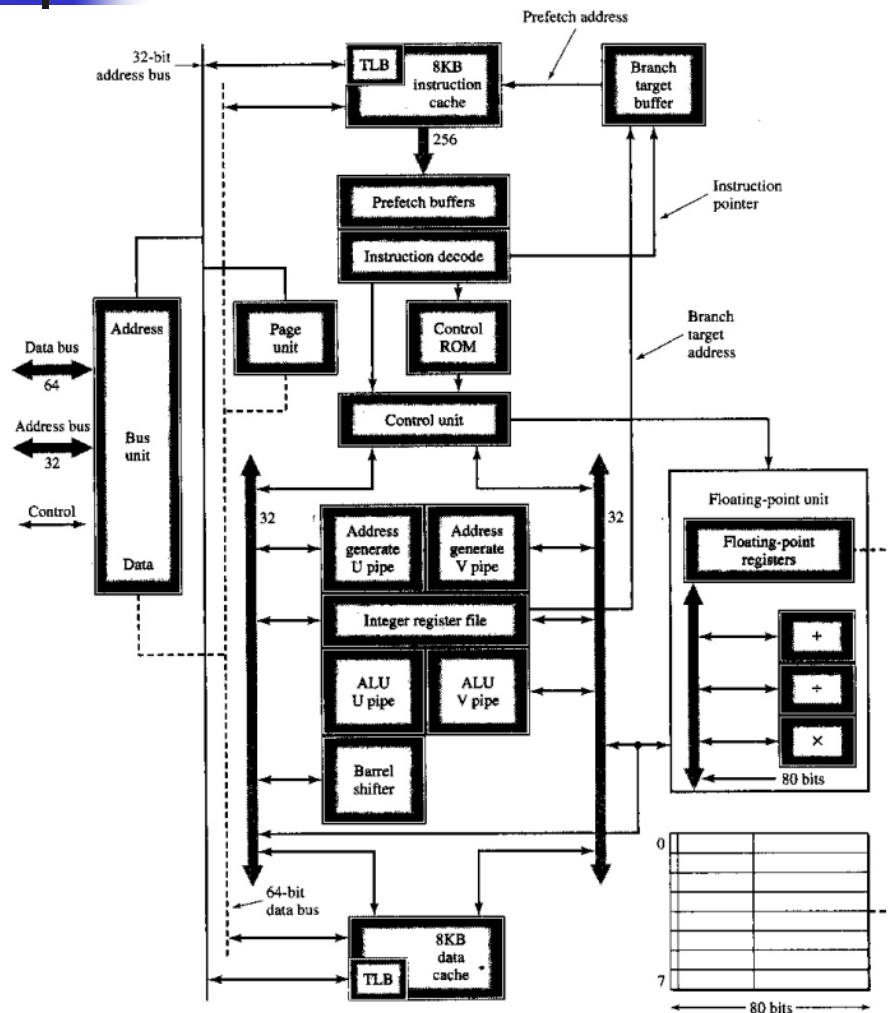


# Mapa pamięci

FIGURE 1-1: PIC16F84A BLOCK DIAGRAM



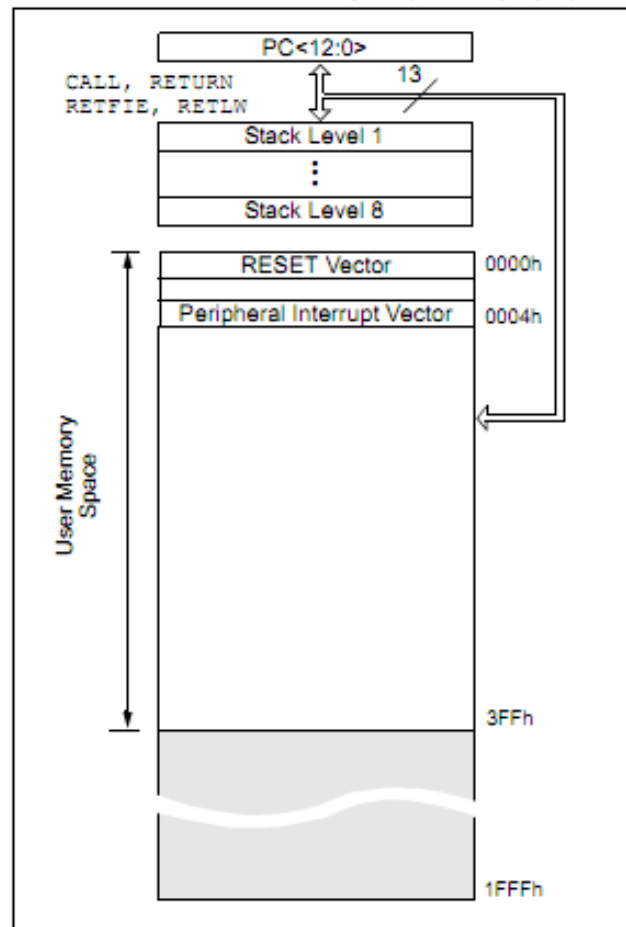
# Mapa pamięci



**Architektura  
obecných  
komputerů**

# Mapa pamięci programu

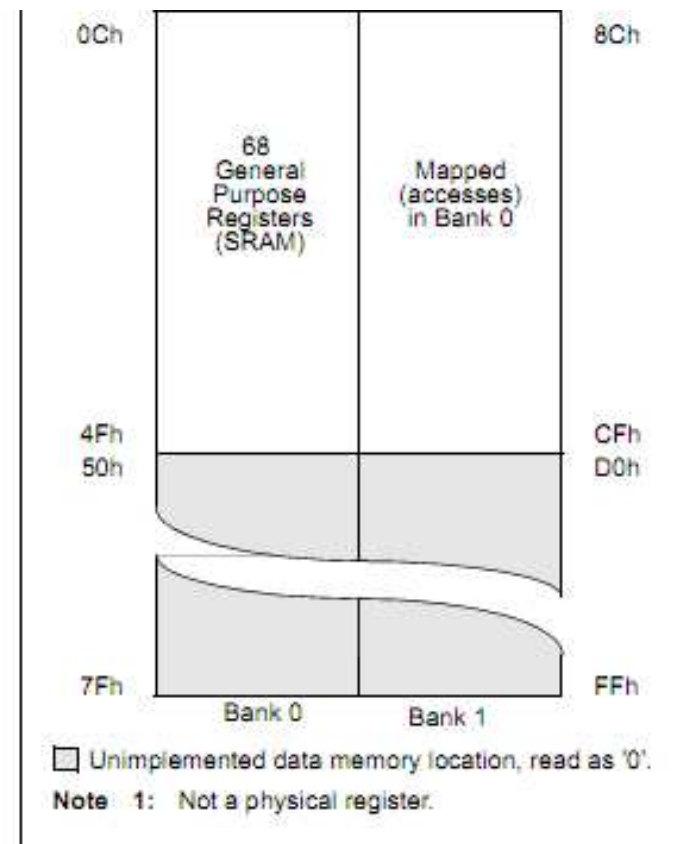
FIGURE 2-1: PROGRAM MEMORY MAP AND STACK - PIC16F84A



# Mapa rejestrów

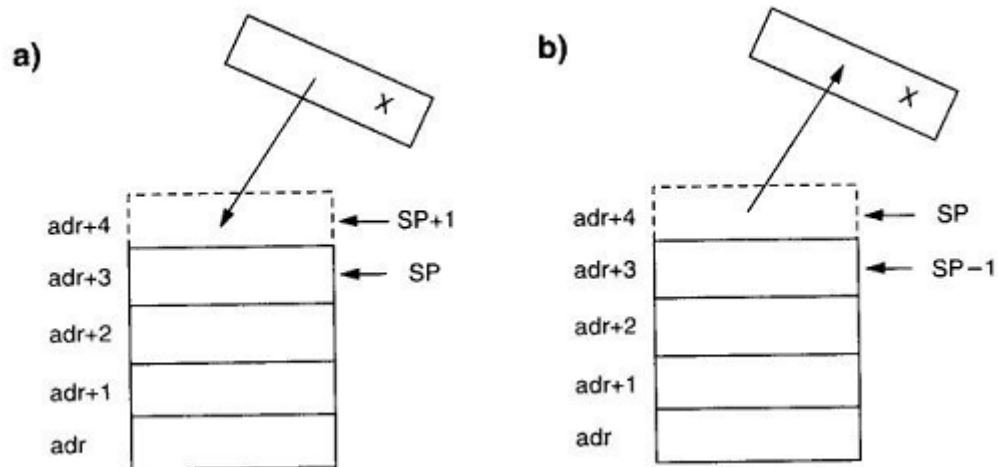
**FIGURE 2-2: REGISTER FILE MAP - PIC16F84A**

File Address		File Address	
00h	Indirect addr. <sup>(1)</sup>	Indirect addr. <sup>(1)</sup>	80h
01h	TMR0	OPTION_REG	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h	—	—	87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2 <sup>(1)</sup>	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh

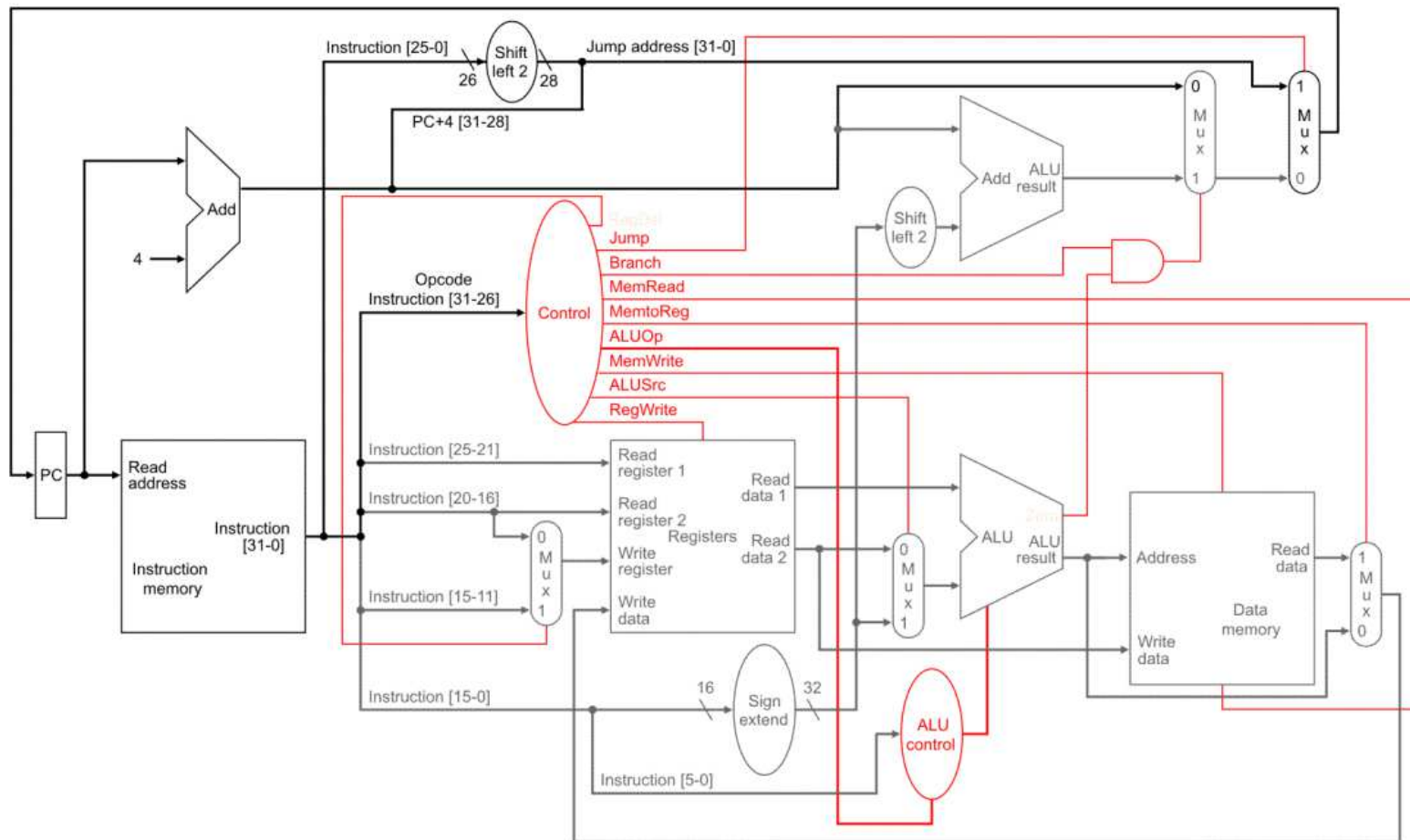


# Stos

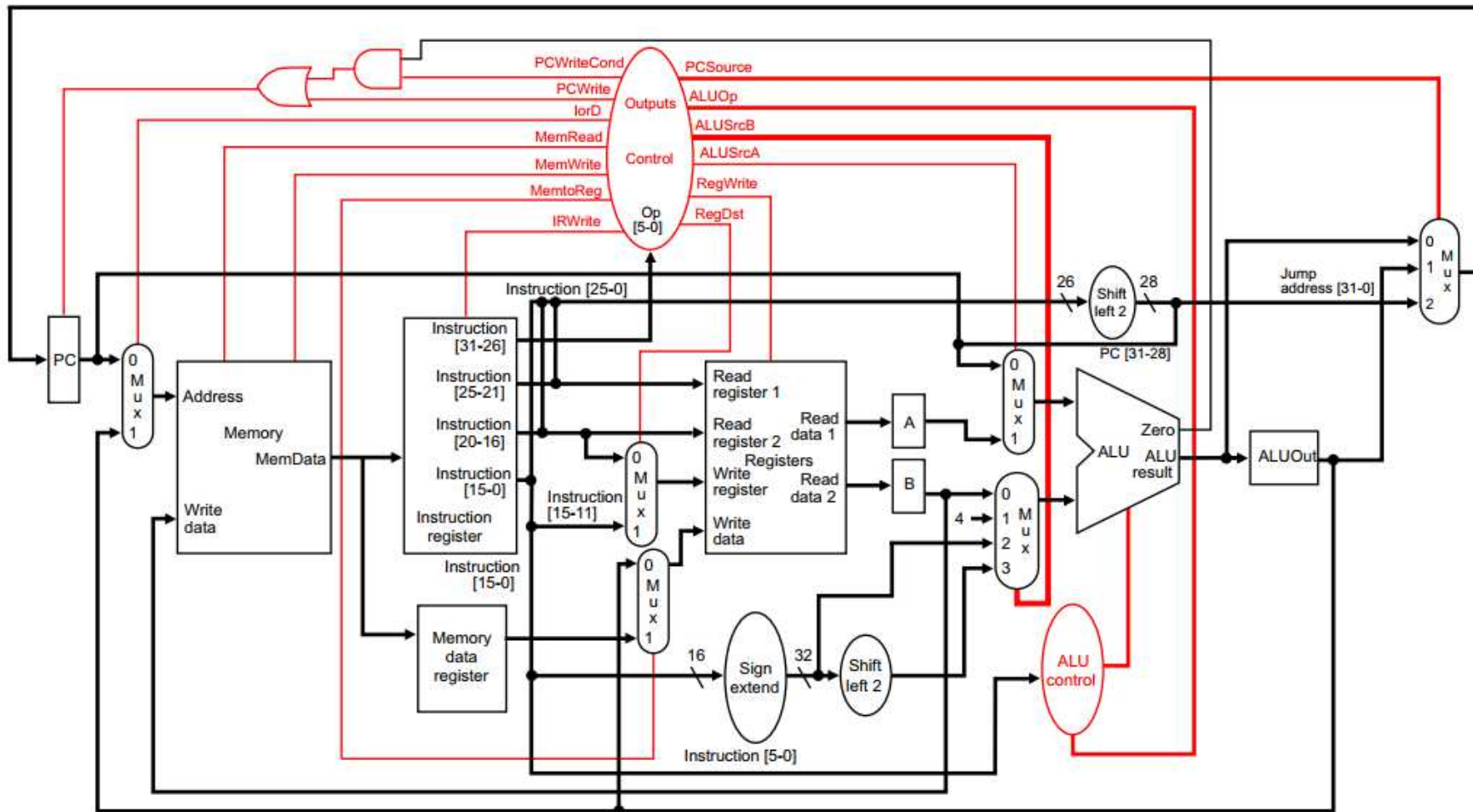
- LIFO
- Rosnący/malejący
- Adres powrotu, kopie rejestrów



# Architektura single cycle



# Architektura multi cycle









# Cykl wykonania instrukcji

Instr. No.	Pipeline Stage						
	IF	ID	EX	MEM	WB		
1	IF	ID	EX	MEM	WB		
2		IF	ID	EX	MEM	WB	
3			IF	ID	EX	MEM	WB
4				IF	ID	EX	MEM
5					IF	ID	EX
Clock Cycle	1	2	3	4	5	6	7

Potoki w  $\mu$ C RISC

- Pentium 3 – 10, P4 – 20, P4HT – 31
- Core i7 – 12-14



# Hazard

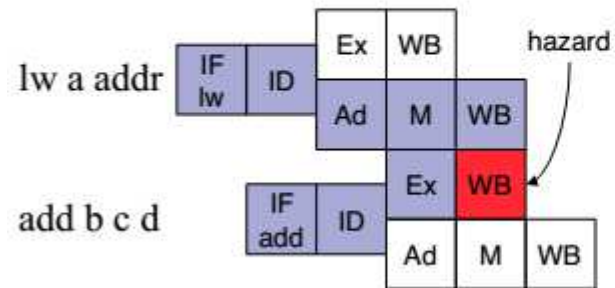
---

- Niepożądane efekty związane z propagacją „sygnału” w warstwie programowej

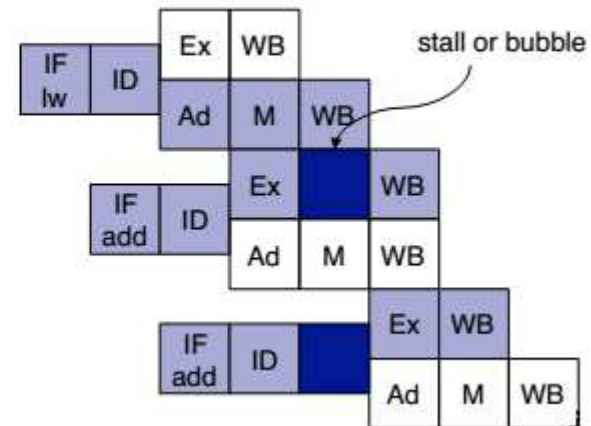


# Hazard

- Structural hazard

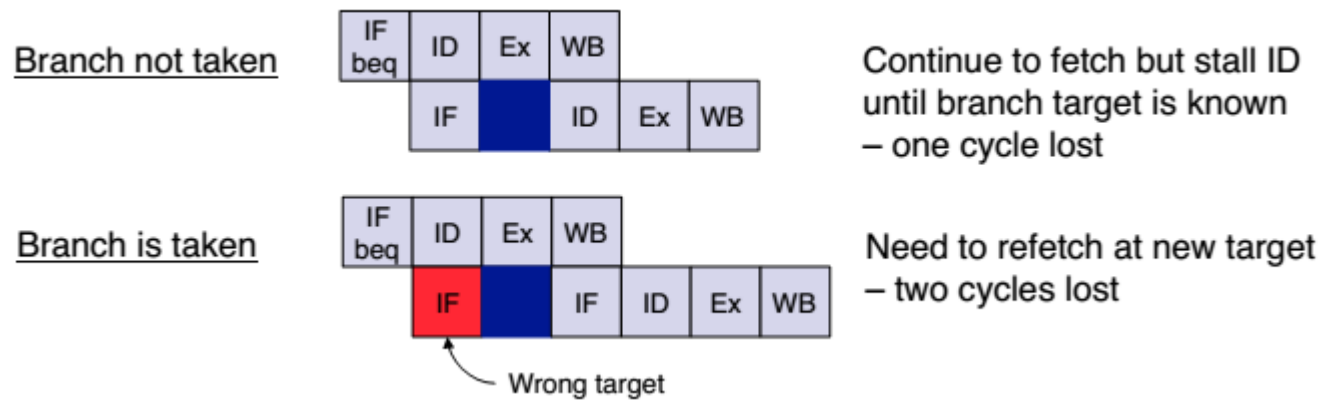


Resolved by stalling the pipeline



# Hazard

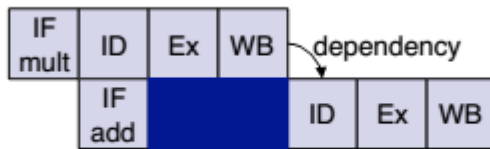
- Control hazard



# Hazard

- Data hazard

mult a b c  
add d a f

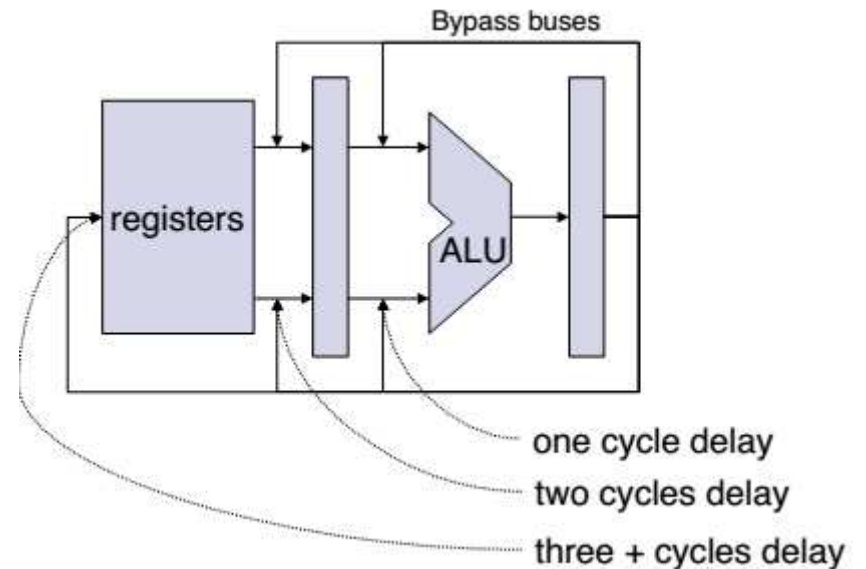


Two bubbles

n.b. register read is in the ID stage



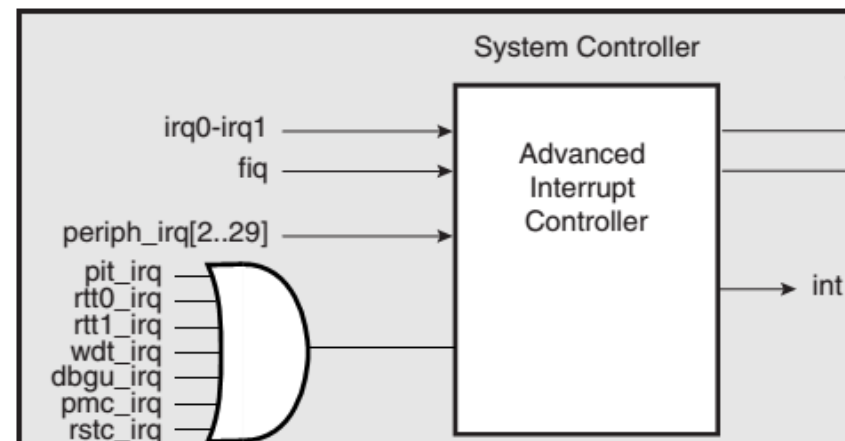
Data bypassed from pipeline register to execution unit



# Przerwania

- Źródła przerw (wewnętrzne, zewnętrzne)
- Sposoby wykrywania (sprzętowe, programowe)
- Priorytety

AT91SAM9263 System Controller Block Diagram





# RISC vs CISC

---

- Krótka lista rozkazów
- Ortogonalność listy rozkazów (ta sama długość)
- Brak rejestrów wyróżnionych do wykonywania konkretnych instrukcji
- Szybkość wykonania instrukcji

Obecnie produkowane procesory Intela z punktu widzenia programisty są widziane jako CISC, ale ich rdzeń jest zgodny z RISC. Rozkazy CISC są rozbijane na mikrorozkazy (ang. microops), które są następnie wykonywane przez szybki blok wykonawczy zgodny z architekturą RISC.