

Technika cyfrowa 2

wykład 3

Architektura procesora typu CISC

J. Biernat: *Architektura komputerów.*

Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2001

- **CISC (Complex Instruction Set Computer)** - procesor:
 - o złożonej liście rozkazów mających różny, zmienny format
 - z małym zestawem rejestrów strukturalnych (dla użytkownika)
 - o rozbudowanych trybach adresowania
- wzrost liczby taktów zegarowych przy wzroście złożoności instrukcji
- trudności z racjonalnym wykorzystaniem zasobów procesora; przy prostych, elementarnych operacjach wykorzystanie niewielkiej części zasobów procesora
- statystyka działania systemów operacyjnych i programów użytkowych wskazuje na częste wykonywanie operacji prostych i rzadkie złożonych
- sterowanie przepływem rozkazów realizowane programowo (mikro-kody); łatwa realizacja sterowania wykonywania złożonych rozkazów wymagających różnych okresów czasu

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

What is "microcode"?

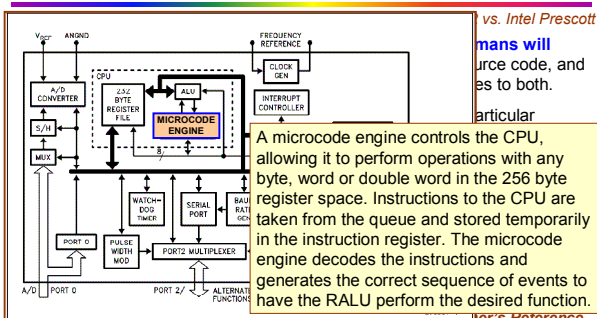


Figure 1-1. 80C196KB Block Diagram

80C196KB User's Guide. INTEL Corporation, November 1990, Order Number: 270651-003

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

C51 - procesor typu CISC

- różny, zmienny format rozkazów:

0010101	r r r	ADD	A,Rn
01111011	i	MOV	@Ri,#dana
AAA00001		ACALL	adr_11
101110101		CJNE	A,adr,rel

- wzrost liczby taktów zegarowych przy wzroście złożoności instrukcji:
 - 1 lub 2 cykle maszynowe dla wszystkich instrukcji
 - 4 cykle maszynowe dla MUL AB oraz DIV AB (standard)
- elementarne operacje wykorzystują tylko wybrane rejestry:
 - uprzywilejowany akumulator A i znacznik przeniesienia C
 - adresowanie pośrednie tylko przez R0 i R1 (z 8 dostępnych rejestrów)

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Architektura procesora typu RISC (1/2)

Zd. Pogoda: *Mikroprocesory RISC rodziny PowerPC.*

Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice 1995

- uproszczoną strukturę mikroprocesora, o zredukowanej liście rozkazów zaproponował John Cocke (IBM Research w Yorktown, New York, 1974) wychodząc z założenia, że 20% dostępnych instrukcji zajmuje aż 80% czasu pracy procesora
- skrót **RISC (Reduced Instruction Set Computer)** zaproponował David Patterson (University of California, Berkeley, 1985)
- obecnie skrót **RISC** tłumaczony jako **Rational Instruction Set Computer**
- idea RISC została wykorzystana w mikroprocesorze SPARC firmy Sun Microsystems; początek powstania tego co jest znane jako technologia MIPS, np. w komputerach Silicon Graphics, procesorze Alpha firmy DEC

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Architektura procesora typu RISC (2/2)

- prosta struktura wewnętrzna
- sterowanie przepływem rozkazów realizowane sprzętowo (brak mikro-kodów, eliminacja mikroprogramowania)
- uproszczenie trybów adresowania, zwykle jednolity format rozkazów (stała długość kodów instrukcji)
- większe możliwości wyboru uniwersalnych rejestrów mikroprocesora
- architektura typu load/store, przetwarzanie danych w oparciu o rejestry nie bezpośrednio na zawartości pamięci
- szybsze działanie → ideałem jest wykonanie każdej instrukcji w jednym cyklu procesora → zmniejszenie liczby taktów zegarowych tworzących cykl maszynowy (przetwarzanie potokowe)
- ułatwienie opracowania, wytworzenia i testowania:
 - nowego mikroprocesora
 - systemu operacyjnego i programu użytkowego
 - kompilatora wysokiego poziomu
- wykonanie operacji równoważnej wymaga większej liczby rozkazów w procesorze RISC niż w procesorze CISC

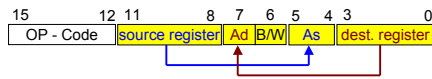
KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

MSP430 - procesor typu RISC

www.ti.com

- 27 basic instructions + 24 emulated instructions
- jednolity format rozkazów (stała długość kodu), np. dla formatu I:



- większe możliwości wyboru uniwersalnych rejestrów mikroprocesora:
 - R0 - licznik rozkazów PC
 - R1 - wskaźnik stosu SP
 - R2 - rejestr statusowy SR / generator stałych CG1
 - R3 - generator stałych CG2
 - R4 .. R15 - rejestry uniwersalne
- ortogonalne tryby adresowania - wszystkie instrukcje ze wszystkimi trybami adresowania i operandami

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

ARM (procesor typu RISC) - Pipeline (1/2)

Up to the **ARM7**, ARM processors have a **3-stage** instruction pipeline:

- **Fetch:** Fetching an instruction from the memory containing the code
- **Decode:** Decoding the instruction and prepare data path control signals for next cycle
- **Execute:** The instruction gets executed on the data path specified and the result is written back to the destination

Higher performance **ARM9** derivatives use a **5-stage** pipeline to compensate for the memory access bottleneck of the 3-stage pipeline:

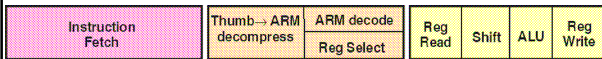
- **Fetch:** Fetch next instruction from memory
- **Decode:** Decode instruction and read register operands
- **Execute:** Execute instruction
- **Data:** Access data memory, if required
- **Write-back:** Write the result of the instruction back to the destination memory location

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

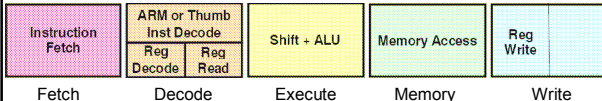
ARM7TDMI / ARM9TDMI - Pipeline (2/2)

ARM7TDMI (ARM7TDMI-S)



Fetch Decode Execute

ARM9TDMI



Fetch Decode Execute Memory Write

- The maximum clock frequency of the ARM9TDMI core is generally in the range 1.8 to 2.2 times the clock frequency of the ARM7TDMI core when compared on the same silicon process.
- Reducing the number of cycles for loads and stores gives a significant improvement in program execution time as typically around 30% of instructions are loads or stores.

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Architektura procesora typu Bit Slice

- każdy blok funkcjonalny tworzą oddzielne układy zwane **segmentami**, np. ALU, rejestry
- istnieje możliwość wzajemnej współpracy (połączenia) segmentów w celu zbudowania procesora o zadanej długości słowa, np. dysponując procesorem 4-bitowym można zbudować procesor 8-, 16-, 24- ...-bitowy
- mikrokod procesora (*microcode*) jest pamiętany w pamięci ROM lub szybkiej pamięci RAM (dynamiczne mikroprogramowanie)
- najpopularniejszym procesorem segmentowym jest seria Am2900 (*Advanced Micro Devices*), np. Am2901:
 - **struktura:** 4-bitowa ALU, 16x4-bitowa pamięć RAM (rejestry), akumulator Q
 - **wykonywane operacje:** dodawanie z przeniesieniem (*addition with carry*), odejmowanie z pożyczką (*subtraction with borrow*), OR, AND, XOR i XNOR
 - **argumenty** zawarte w: RAM, akumulatorze, zewnętrznych wejściach, domyślna wartość 0,
 - **wynik operacji:** przeniesienie dla innych segmentów (*carry-out flag*), znak (*sign flag*), przekroczenie zakresu (*overflow flag*), wynik zerowy (*zero flag*)...

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Optymalizacja, przetwarzanie danych

Optymalizacja kompilatorów:

- rozkazy wykonywane przez procesor realizują funkcje, które najczęściej powtarzają się podczas kompilacji
- lista rozkazów dostosowana do potrzeb kompilatorów, a nie własności procesora
- założenia przydatne przy projektowaniu nowych konstrukcji procesorów

Sposób przetwarzania danych - obciążenie rejestrów procesora i pamięci:

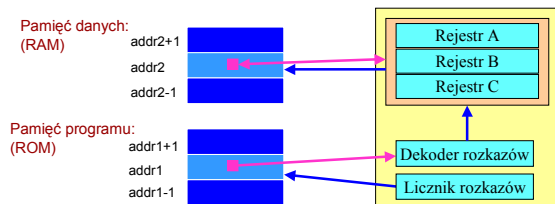
- **register-to-register** → przestrzeni roboczą są rejestry ogólnego przeznaczenia:
 - **accumulator-based** CPU, z akumulatorem związana jest większość instrukcji arytmetyczno-logicznych i wymiany danych
 - **register-based** CPU (register-to-register), rolę akumulatora pełnią rejestry ogólnego przeznaczenia
- **memory-to-memory** → bloki pamięci traktowane są jako przestrzeń roboczą, zastępująca rejestry ogólnego przeznaczenia

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Tryby adresowania - struktura procesora

- licznik rozkazów PC adresujący pamięć kodu programu (ROM)
- dekodery rozkazów
- 3 wewnętrzne rejestry: A, B i C
- mechanizmy dostępu do pamięci danych (RAM) – adresowanie bezpośrednie i pośrednie (za pośrednictwem rejestru: A, B lub C)

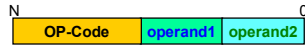


KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Tryby adresowania - właściwe

- właściwe** (*inherent, implied*) - wszystkie informacje potrzebne do wykonania rozkazu zawiera kod operacyjny instrukcji (opcode)



Pamięć danych:



Pamięć programu:



KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

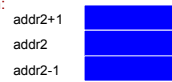
opracowanie: Andrzej Stasiak

Tryby adresowania - natychmiastowe

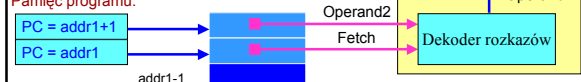
- natychmiastowe** (*immediate*) - operand (operandy) zawarty w kolejnym bajcie lub bajtach stanowiących rozkaz



Pamięć danych:



Pamięć programu:

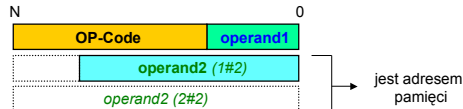


KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Tryby adresowania - bezpośrednie

- bezpośrednie** (*direct*) - adres operandu umieszczony w rozkazie, w bajcie lub bajtach tworzących rozkaz



Pamięć danych:



Pamięć programu:

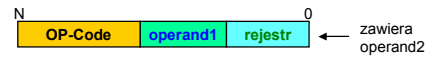


KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

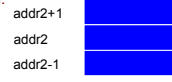
opracowanie: Andrzej Stasiak

Tryby adresowania - rejestrowe

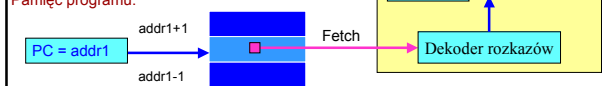
- rejestrowe** (*register*), **rejestrowe bezpośrednie** (*register direct*) - operand zawarty w rejestrze; adres rejestru stanowi część rozkazu



Pamięć danych:



Pamięć programu:



KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

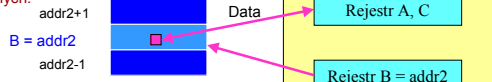
opracowanie: Andrzej Stasiak

Tryby adresowania - pośrednie

- rejestrowe pośrednie** (*register indirect*) - w rejestrze adres operandu (argumentu) w pamięci; adres rejestru stanowi część rozkazu



Pamięć danych:



Pamięć programu:



KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Tryby adresowania - wskaźnikowe

- wskaźnikowe** (*pointer*) - adres argumentu w dedykowanym rejestrze; odmiana adresowania pośredniego
- indeksowe** (*indexed*) - w rejestrze indeksowym adres argumentu w pamięci
- indeksowe bez przesunięcia** (*indexed, no offset*) - adres w rejestrze indeksowym, brak przesunięcia (Motorola)



Pamięć danych:



Pamięć programu:

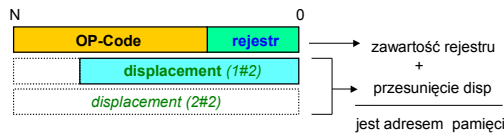


KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Tryby adresowania - indeksowe (1/2)

- absolutne indeksowe** (*absolute indexed*) - adres argumentu jest sumą zawartości rejestru indeksowego i 16-bitowego przesunięcia
- indeksowe z 8-bitowym przesunięciem** (*indexed, 8-bit offset*) - adres jest sumą zawartości rejestru indeksowego i 8-bitowego przesunięcia (Motorola)
- indeksowe z 16-bitowym przesunięciem** (*indexed, 16-bit offset*) - adres jest sumą zawartości rejestru indeksowego i 16-bitowego przesunięcia
- rejestrowe pośrednie z przesunięciem** (*register indirect with displacement lub absolute offset indirect*) - adres pamięci jest sumą zawartości rejestru i przesunięcia; adres rejestru i przesunięcie stanowią część rozkazu

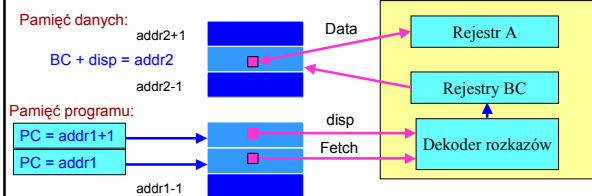


KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Tryby adresowania - indeksowe (2/2)

- absolutne indeksowe** (*absolute indexed*) - adres argumentu jest sumą zawartości rejestru indeksowego i 16-bitowego przesunięcia
- indeksowe z 8-bitowym przesunięciem** (*indexed, 8-bit offset*) - adres jest sumą zawartości rejestru indeksowego i 8-bitowego przesunięcia (Motorola)
- indeksowe z 16-bitowym przesunięciem** (*indexed, 16-bit offset*) - adres jest sumą zawartości rejestru indeksowego i 16-bitowego przesunięcia
- rejestrowe pośrednie z przesunięciem** (*register indirect with displacement lub absolute offset indirect*) - adres pamięci jest sumą zawartości rejestru i przesunięcia; adres rejestru i przesunięcie stanowią część rozkazu

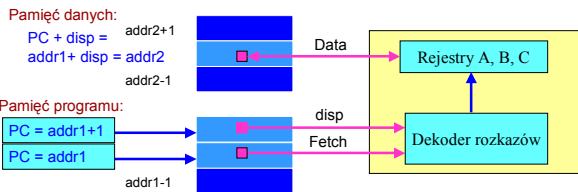
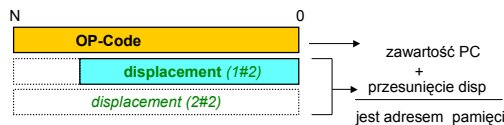


KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Tryby adresowania - względne

- względne** (*relative lub relative direct*) - adres argumentu jest sumą bieżącej zawartości licznika rozkazów PC i 8- lub 16-bitowego przesunięcia

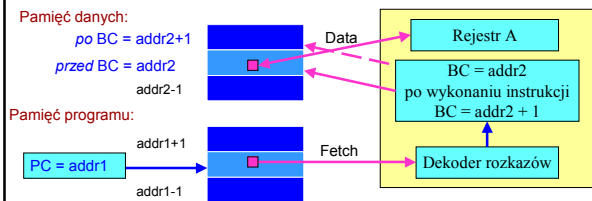
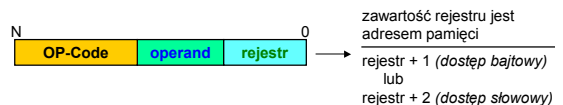


KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Tryby adresowania - pośrednie + INC

- rejestrowe pośrednie z autoinkrementacją** (*register indirect, autoincrement / with post-increment*) - adres argumentu w rejestrze; adres rejestru stanowi część rozkazu, po wykonaniu rozkazu autoinkrementacja zawartości rejestru



KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

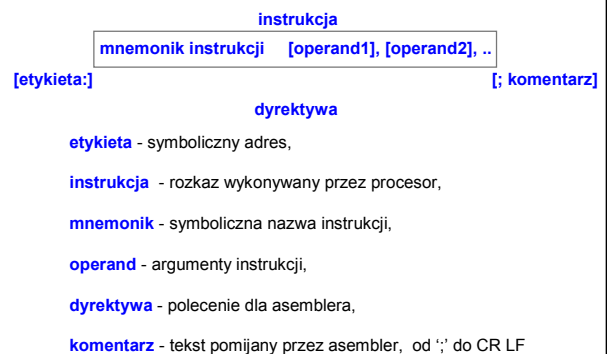
Tryby adresowania - inne ?

- względne** (*relative lub relative direct*) - adres argumentu jest sumą bieżącej zawartości licznika rozkazów PC i 8- lub 16-bitowego przesunięcia
- względne pośrednie** (*relative indirect*) - adres argumentu jest sumą bieżącej zawartości licznika rozkazów i rejestru lub rejestrów
- indeksowo-rejestrowo-pośrednie** (*indirect-register-indexed*) - adres jest sumą zawartości rejestru indeksowego i rejestru ogólnego przeznaczenia
- względne indeksowe** (*relative indexed*) - adres argumentu jest sumą bieżącej zawartości licznika rozkazów, rejestru indeksowego i 16-bitowego przesunięcia
- względne pośrednie z przesunięciem** (*relative offset indirect*) - adres argumentu jest sumą bieżącej zawartości licznika rozkazów, rejestru (rejestrów) i 16-bitowego przesunięcia
- bitowe** (*bit*) - adres bitu zawarty w kodzie operacyjnym rozkazu lub w kolejnym bajcie lub bajtach stanowiących rozkaz

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Linia kodu programu

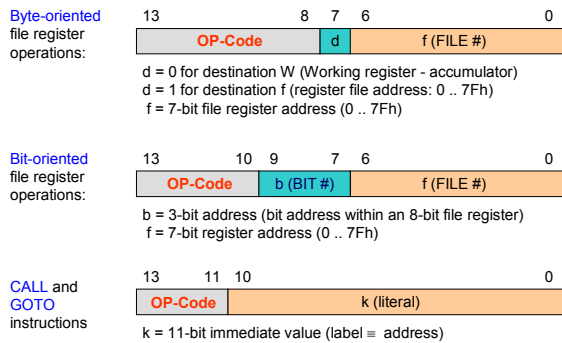


KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Format instrukcji - PIC

PIC16C77x (Microchip Technology Inc.) - stała długość instrukcji równa 14-bitów



KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Format instrukcji - C51 (1/3)

rozkazy 1-bajtowe, 2-bajtowe i 3-bajtowe

kod rozkazu bez argumentów lub z pojedynczym, ukrytym argumentem:



kod rozkazu i argumenty:

→ 1 bit dla adresowania pośredniego:



→ 3 bity dla adresowania rejestrowego:



KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

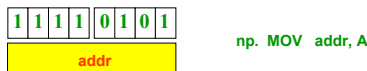
opracowanie: Andrzej Stasiak

Format instrukcji - C51 (2/3)

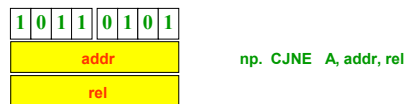
rozkazy 1-bajtowe, 2-bajtowe i 3-bajtowe

kod rozkazu i argumenty:

→ 2 bajty dla adresowania bezpośredniego:



→ 3 bajty dla dwóch 1-bajtowych argumentów:



KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

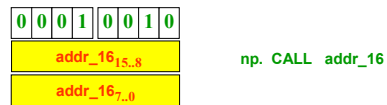
opracowanie: Andrzej Stasiak

Format instrukcji - C51 (3/3)

rozkazy 1-bajtowe, 2-bajtowe i 3-bajtowe

kod rozkazu i argumenty:

→ 3 bajty dla jednego, 2-bajтового argumentu:



Format rozkazu wykorzystujący różne podane kombinacje:



KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

PREFIX w kodach rozkazów C51

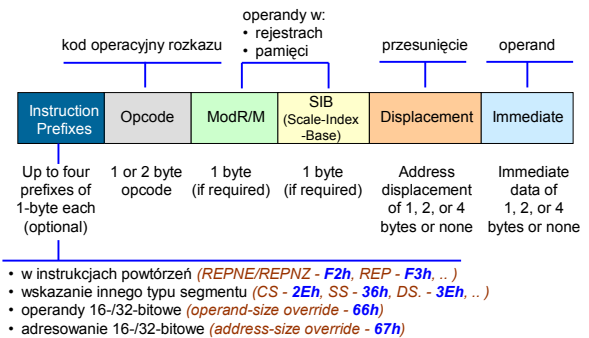
	x0	x1	x2	x3	x5
0x	NOP	AJMP addr11	LJMP addr16	RR A	INC addr
1x	JBC bit, rel	ACALL addr11	LCALL addr16	RRC A	DEC addr
2x	JB bit, rel	AJMP addr11	RET	RL A	ADD A, addr
Ax	ORL C, /bit	AJMP addr11	MOV C, bit	INC DPTR	OPCODE PREFIX OPCODE PREFIX = 0A5h dodatkowe instrukcje np. 80C51MX (Philips) C251 (Intel/Atmel)
Dx	POP addr	ACALL addr11	LJMP addr16	RR A	
Ex	MOVX A, @DPTR	AJMP addr11	MOVX A, @Ri		
Fx	MOVX @DPTR, A	ACALL addr11	MOVX @Ri, A		
					Brak instrukcji: dla CODE do 2KB bez P0 i P2

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Format instrukcji - Pentium

IA-32 Intel® Architecture Software Developer's Manual. Volume 2: Instruction Set Reference



KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Data Transfer (Load or Store) (1/2)

Single Register Data Transfer:

LDR / STR	Word
LDRB / STRB	Byte
LDRH / STRH	Halfword
LDRSB	Signed byte load
LDRSH	Signed halfword load

Multiple Registers Data Transfer:

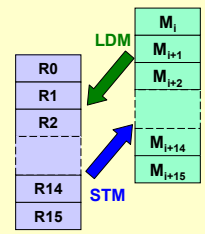
LDMIA / STMIA	increment after
LDMIB / STMIB	increment before
LDMDA / STMDA	decrement after
LDMDB / STMDB	decrement before

Data Transfer (Load or Store) (2/2)

Load / Store Multiple instructions (LDM / STM)

Single Register Data Transfer:

Whole register bank or a subset copied to memory or restored with single instruction



Multiple Registers Data Transfer:

LDMIA / STMIA	increment after
LDMIB / STMIB	increment before
LDMDA / STMDA	decrement after
LDMDB / STMDB	decrement before

ARM - Jazelle® Technology

The Architecture for the Digital World

- Jazelle cores can also execute Java bytecode. ARM Jazelle® technology for Java™ acceleration delivers an unparalleled combination of Java performance and the world's leading 32-bit embedded RISC architecture - giving platform developers the freedom to run Java applications alongside established OS, middleware and application code on a single processor. The single-processor solution offers higher performance, lower system cost and lower power than coprocessors and dual-processor (dedicated Java processor and native applications processor) solutions.



Jazelle enabled



Jazelle disabled

Struktura procesora

- R0 - Program Counter PC
- R1 - Stack Pointer SP
- R2 - Status Register SR / Constant Generator CG1 (+4, +8)

RESERVED	V	SCG1	SCG0	OSC	CPU	GIE	N	Z	C
		1	0	Off	Off				

most used constant: +8 +4 +2 +1

- R3 - Constant Generator CG2 (0, +1, +2, -1)
- R4 .. R15 - User working Registers

Typy rozkazów

Mode	Example	Description
Register	MOV Rn, Rm	Register contents are operands
Indexed	MOV X(Rn), Y(Rm)	(Rn + X) is the operand's address
Symbolic	MOV EDE, TONI	(PC + X) is the operand's address
Indirect	MOV @Rn, Rm	Rn points to the operand
Indirect autoincrement	MOV @Rn+, Rm	Rn points to the operand, post-incr.
Immediate	MOV #45, TONI	The word after the instruction contains the immediate constant

Rozkazy bazowe i emulowane

MOV - move source to destination: scr ⇒ dst

MOV ← BR	Branch to destination:	MOV Rn, PC
CLR	Clear destination:	MOV #0, dst
NOP	No operation:	MOV #0, #0
POP	Pop word from stack to destination:	MOV @SP+ dst
RET	Return from subroutine:	MOV @SP+, PC

ADD - add source operand to the destination operand: scr + dst ⇒ dst

ADD ← INC	Increment destination:	ADD #1, dst
INCD	Double-Increment destination:	ADD #2, dst
RLA	Rotate left arithmetically:	ADD dst, dst

BIC - clear bits in destination: .NOT. scr .AND. dst ⇒ dst

BIC ← CLRC	Clear Carry bit:	BIC #1, SR
CLRZ	Clear Zero bit:	BIC #2, SR
CLRN	Clear Negative bit:	BIC #4, SR
DINT	Disable general interrupts:	BIC #8, SR

Rejestry procesora ARM

- Linear address space, 32-bit address space ($2^{32} = 4 \text{ Gbyte}$)
- The ARM processor has a total of 37 registers:
 - 31 general-purpose registers, including a program counter. These registers are 32 bits wide and are visible:
 - the unbanked registers, R0 – R7,
 - the banked registers, R8 – R14,
 - register R15 is a Program Counter PC.
 - 6 status registers. These registers are also 32 bit wide, but only 12 of the 32 bits are allocated or need to be implement.
- Registers are arranged in partially overlapping banks, with a different register bank for each processor mode. At any time, 15 general-purpose registers (R0 to R14), one or two status registers and the program counter are visible:

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Dlaczego MCS[®]51 ???

Acer Labs	Actel	Aeroflex UTM
Analog Devices	AnchorChips	Atmel W & M
Cast Inc.	Chipcon	CML Microcircuits
Cybernetic Micro Systems	Cybra Tech.	Cygnal Integrat. Prod.
Cypress Semiconductor	Daewoo	Dallas Semiconductor
Dolphin	Domosys	Generic Genesis
Microchip	Goal Semiconductor	Honeywell
Hynix Semiconductor	Hyundai	Infineon InnovASIC
Intel	ISSI	Maxim Integrated
Product	Mentor Graphics Co.	Micronas
MXIC	Myson Technology	Nordic VLSI
OKI	Oregano Systems	Philips
Sanyo	Sharp	Siemens
Siliconians	Silicon Laboratories	SMSC
SST	ST Microelectronics	Synopsys
TDK	Temec	Texas Instruments
Triscend	Winbond	51. ???????????

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

MCS[®]51 na stronach www

- www.8052.com
- www.atmel.com
- www.cygnal.com / www.silabs.com
- www.infineon.com
- www.intel.com
- www.maxim-ic.com
- www.semiconductors.philips.com
- www.st.com
- www.ti.com

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

C51 - Typy rozkazów, adresowanie

- wymiany danych (*data transfer*)
- logiczne (*logical operations*)
- arytmetyczne (*arithmetic operations*)
- manipulacji bitowych (*boolean variable manipulation*)
- skoków i wywołań podprogramów (*program branching*)
- rejestrowe:
 - MOV A, R1 ;A ← R1, nazwa rejestru
- bezpośrednie:
 - MOV A, 1 ;A ← (1), adres pamięci
- pośrednie:
 - MOV A, @R1 ;A ← (R1), adres pamięci w R1
- natychmiastowe:
 - MOV A, #3 ;A ← 3, wartość
- indeksowo-rejestrowe-pośrednie:
 - MOVC A, @A+DPTR ;A ← (A+DPTR), adres pamięci w A+DPTR

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

C51 - Skróty, symbole, oznaczenia

- Rn (R0..R7)** - nazwa rejestru w bieżącym banku rejestrów,
- addr** - adres wewnętrznej pamięci RAM lub rejestrów specjalnych SFR,
- @Ri (R0, R1)** - rejestr adresujący pośrednio pamięć RAM,
- #data** - zmienna 8-bitowa,
- #data_16** - zmienna 16-bitowa, *tylko w instrukcji MOV DPTR, #data_16*
- addr_11** - 11-bitowy adres w pamięci kodu programu,
- addr_16** - 16-bitowy adres w pamięci kodu programu,
- rel** - 8-bitowa wartość w kodzie uzupełnienia do 2 (U2) dodawana do licznika rozkazów PC,
- bit** - adres bitu w wewnętrznej pamięci RAM lub w rejestrach specjalnych SFR

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Lista rozkazów mikroprocesora 8051. Instrukcje wymiany danych

Mnemonic	Operacja	Struktura bajtów	Cykle	Znaczniki
MOV A,Rn	A ← Rn	[1 1 1 0, 1 r r r]	1	P
MOV A,addr	A ← (addr)	[1 1 1 0, 0 1 0 1]	1	P
MOV A,@Ri	A ← (Ri)	[1 1 1 0, 0 1 1 1]	1	P
MOV A,#data	A ← data	[0 1 1 1, 0 1 0 0]	1	P
MOV Rn,A	Rn ← A	[1 1 1 1, 1 r r r]	1	—
MOV Rn,addr	Rn ← (addr)	[1 0 1 0, 1 r r r]	2	—
MOV Rn,#data	Rn ← data	[0 1 1 1, 1 r r r]	1	—
MOV addr,A	(addr) ← A	[1 1 1 1, 0 1 0 1]	1	—
MOV addr,Rn	(addr) ← Rn	[1 0 0 0, 1 r r r]	2	—
MOV addr,addr1	(addr) ← (addr1)	[1 0 0 0, 0 1 0 1]	2	—
MOV addr,@Ri	(addr) ← (Ri)	[1 0 0 0, 0 1 1 1]	2	—
MOV addr,#data	(addr) ← data	[0 1 1 1, 0 1 0 1]	2	—
MOV @Ri,A	(Ri) ← A	[1 1 1 1, 0 1 1 1]	1	—
MOV @Ri,addr	(Ri) ← (addr)	[1 0 1 0, 0 1 1 1]	2	—

Oznaczenia: **r** - rejestr R0..R7 wybranego banku rejestrów (rr=000 dla R0, rr=111 dla R7)

Ri - rejestr R0 (i=0) lub R1 (i=1)

Rn - rejestr R0..R7 (n=0 dla R0, n=7 dla R7)

data - zmienna 8-bitowa

data_16 - zmienna 16-bitowa

WWW.KMEIF.PWR.WROC.PL

- Dydaktyka → Technika cyfrowa i mikroprocesorowa
- Materiały pomocnicze do zajęć laboratoryjnych
- Rysunki *.gif (rozdzielczość 75 dpi oraz 300dpi)

MOVX @DPTR,A	(DPTR) ← A	[1 1 1 1, 0 0 0 0]	2	—	zawartość
--------------	------------	--------------------	---	---	-----------

Pierwszy program - plik *.a51 (1/5)

;Przykład programu

```
CSEG AT 0 ; pierwsza instrukcja rozpoczyna się
; od adresu 0
```

Początek:

```
MOV R7, #248 ; 1 R7 ← 248 = 0F8h
```

Skok1:

```
DJNZ R7, Skok1 ; 2 R7 ← R7-1
; jeśli R7 ≠ 0 to wykonaj instrukcję
; o etykiecie Skok1
; jeśli R7 = 0 to wykonaj następną instrukcję
```

Negacja:

```
CPL P1.1 ; 1 P1.1 ← NOT (P1.1)
LJMP Początek ; 2 rozpocznij wykonywanie instrukcji
; od etykiety Początek
```

```
END ; koniec pliku źródłowego
```

KATEDRA METROLOGII ELEKTRONICZNEJ | FOTONICZNEJ

opracowanie: Andrzej Stasiak

Pierwszy program - plik *.a51 (2/5)

Przykład programu

```
CSEG AT 0
```

etykieta

Początek:

```
MOV R7, #80h
```

Skok1:

```
NOP
```

```
DJNZ R7, Skok1
```

mnemoniki

operandy

```
CPL P1.1
```

```
LJMP Początek
```

dyrektywa

```
END
```

komentarze

```
; pierwsza instrukcja rozpoczyna się
; od adresu 0

; 1 R7 ← 80h=128

; 1
; 2 R7 ← R7-1
; jeśli R7 ≠ 0 to wykonaj instrukcję
; o etykiecie Skok1
; jeśli R7 = 0 to wykonaj następną instrukcję

; 1 P1.1 ← NOT (P1.1)
; 2 rozpocznij wykonywanie instrukcji
; od etykiety Początek

; koniec pliku źródłowego
```

KATEDRA METROLOGII ELEKTRONICZNEJ | FOTONICZNEJ

opracowanie: Andrzej Stasiak

Pierwszy program - plik *.lst (3/5)

A51 MACRO ASSEMBLER TEST1

02/15/2004 19:02:11 PAGE 1

MACRO ASSEMBLER A51 V6.10

OBJECT MODULE PLACED IN .test1.OBJ

ASSEMBLER INVOKED BY: E:\KEIL\C51\BIN\A51.EXE .test1.a51 SET(SMALL) DEBUG EP

LOC	OBJ	LINE	SOURCE
		1	;Przykład programu
		2	
		3	CSEG AT 0 ; pierwsza instrukcja rozpoczyna się
		4	; od adresu 0
0000		5	Początek:
0000 7FF8		6	MOV R7, #248 ; 1 R7 ← 248 = 0F8h
0002		7	Skok1:
0002 DFFE		8	DJNZ R7, Skok1 ; 2 R7 ← R7-1
		9	; jeśli R7 <> 0 to wykonaj instrukcję
		10	; o etykiecie Skok1
		11	; jeśli R7 = 0 to wykonaj następną instrukcję
0004		12	Negacja:
0004 B291		13	CPL P1.1 ; 1 P1.1 ← NOT (P1.1)
0006 020000		14	LJMP Początek ; 2 rozpocznij wykonywanie instrukcji
		15	; od etykiety Początek
		16	END ; koniec pliku źródłowego

KATEDRA METROLOGII ELEKTRONICZNEJ | FOTONICZNEJ

opracowanie: Andrzej Stasiak

Pierwszy program - plik *.m51 (4/5)

BL51 BANKED LINKER/LOCATER V4.03

02/15/2004 19:02:12 PAGE 1

BL51 BANKED LINKER/LOCATER V4.03, INVOKED BY:

E:\KEIL\C51\BIN\BL51.EXE test1.obj TO test1

INPUT MODULES INCLUDED:
test1.obj (TEST1)

LINK MAP OF MODULE: test1 (TEST1)

TYPE	BASE	LENGTH	RELOCATION	SEGMENT NAME
***** DATA MEMORY *****				
REG	0000H	0008H	ABSOLUTE	"REG BANK 0"
***** CODE MEMORY *****				
CODE	0000H	0009H	ABSOLUTE	

LINK/LOCATE RUN COMPLETE. 0 WARNING(S), 0 ERROR(S)

KATEDRA METROLOGII ELEKTRONICZNEJ | FOTONICZNEJ

opracowanie: Andrzej Stasiak

Pierwszy program - pamięć programu (5/5)

Program
memory
address:

Address	Hex	Instruction	Op1	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31
Początek: 0000	7F	MOV R7, #248	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0001	F8																																
Skok1: 0002	DF	DJNZ R7, Skok1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0003	FE																																
0004	B2	CPL P1.1	1	0	1	1	0	0	1	0																							
0005	91																																
0006	02	LJMP Początek	0	0	0	0	0	0	0	1	0																						
0007	00																																
0008	00																																
0009	xx																																

KATEDRA METROLOGII ELEKTRONICZNEJ | FOTONICZNEJ

opracowanie: Andrzej Stasiak

IDE µVision2 (1/7)

When you use

the [Keil Software Tools](#),

the project development cycle is roughly the same as it is for any other software development project:

1. Create a project to select the device and the tool settings
2. Create source files in assembly or C
3. Build your application with project manager
4. Correct errors in source files
5. Test linked application

KATEDRA METROLOGII ELEKTRONICZNEJ | FOTONICZNEJ

opracowanie: Andrzej Stasiak

