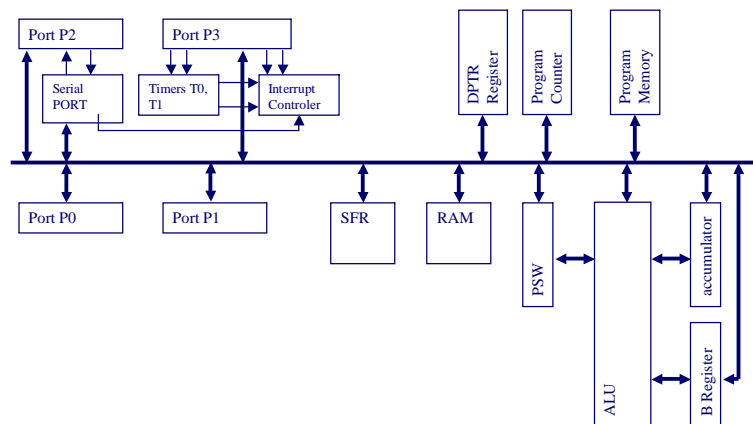
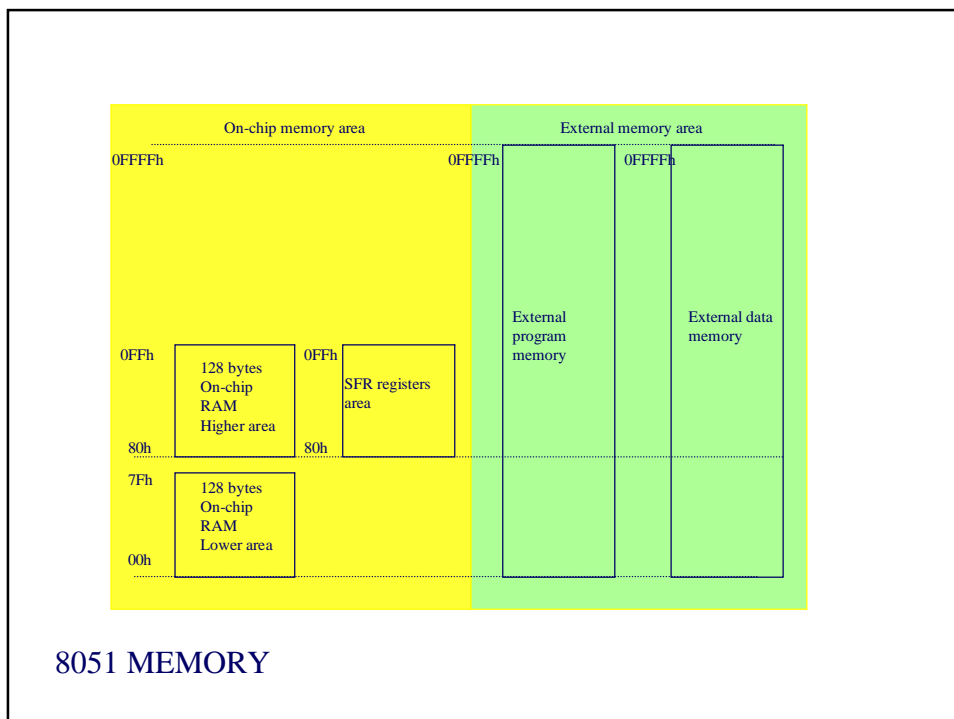
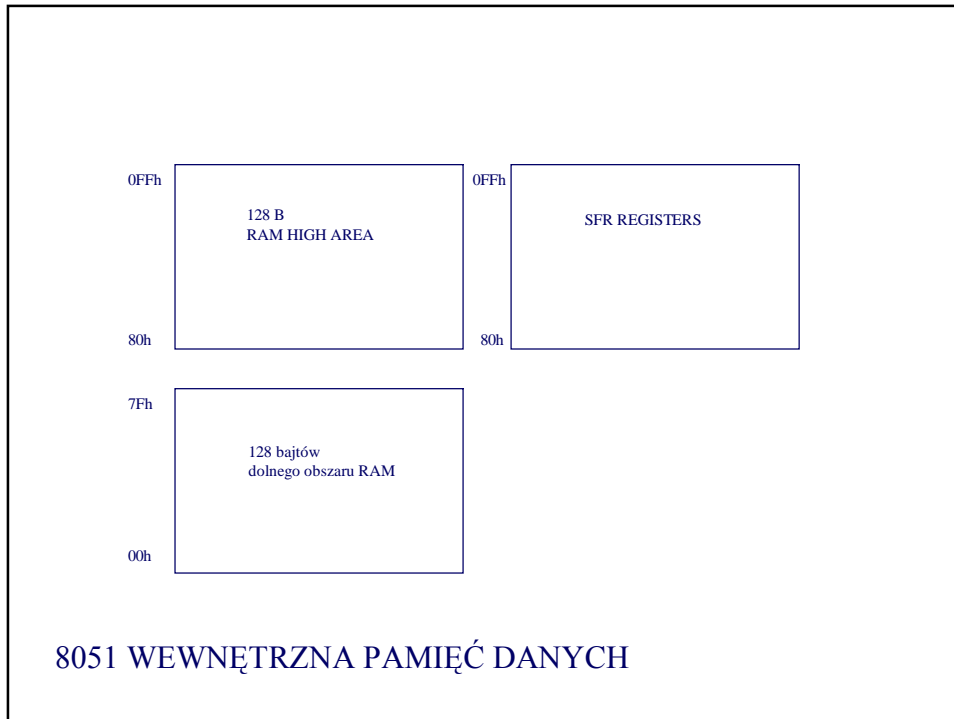
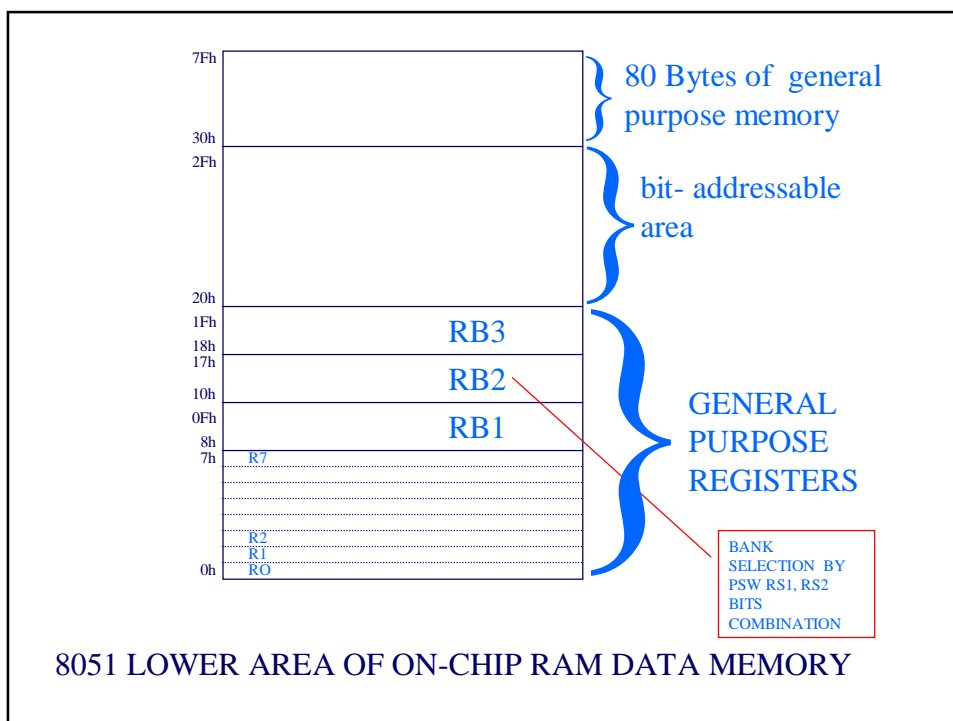
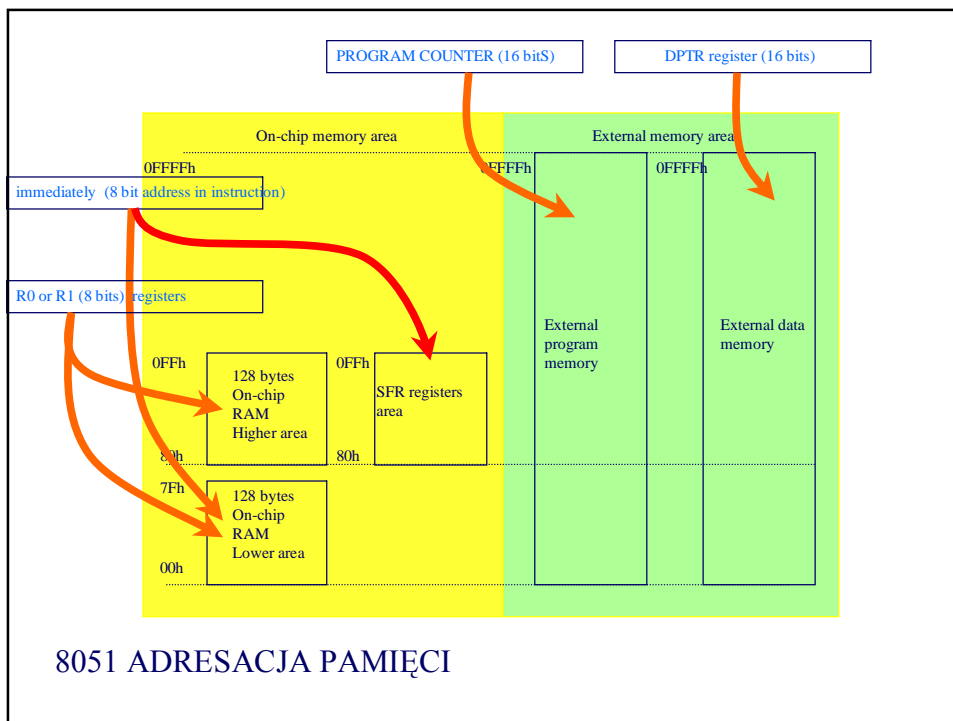


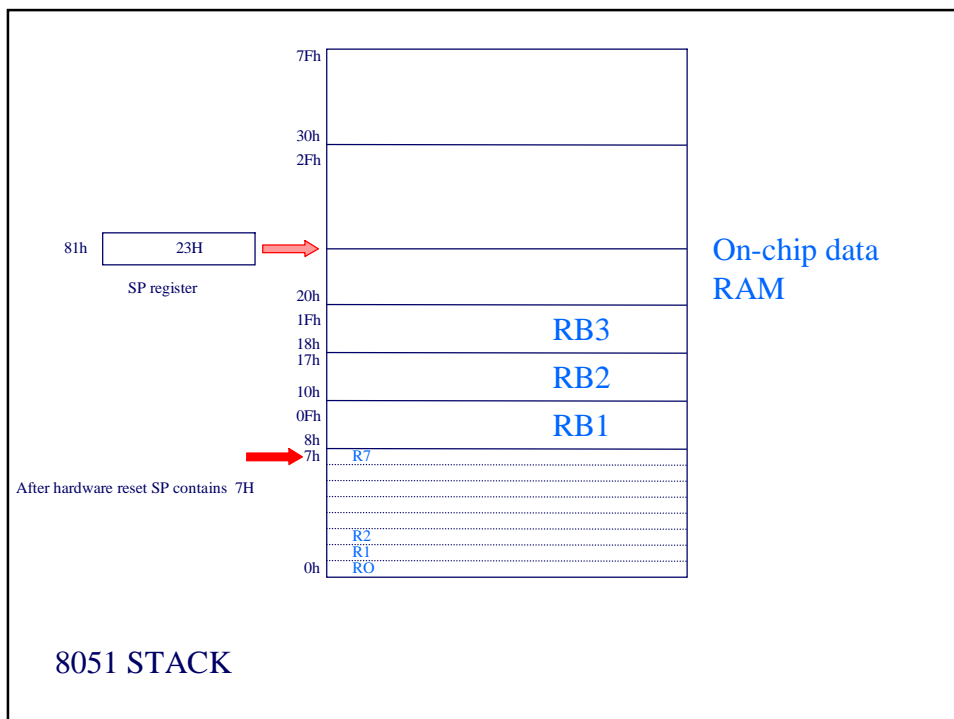
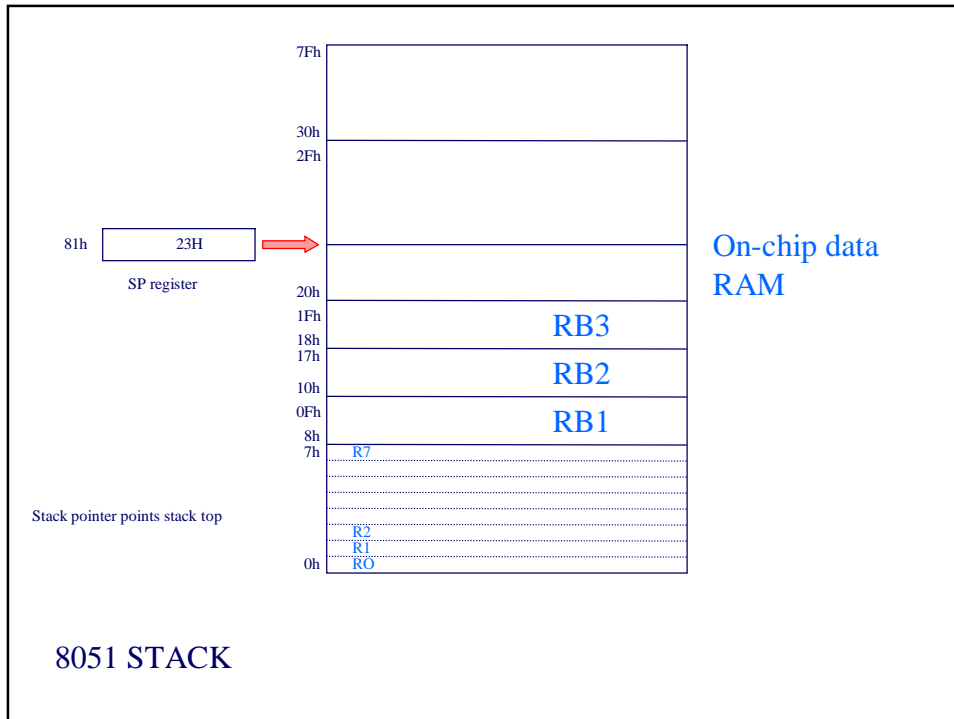
# 8051

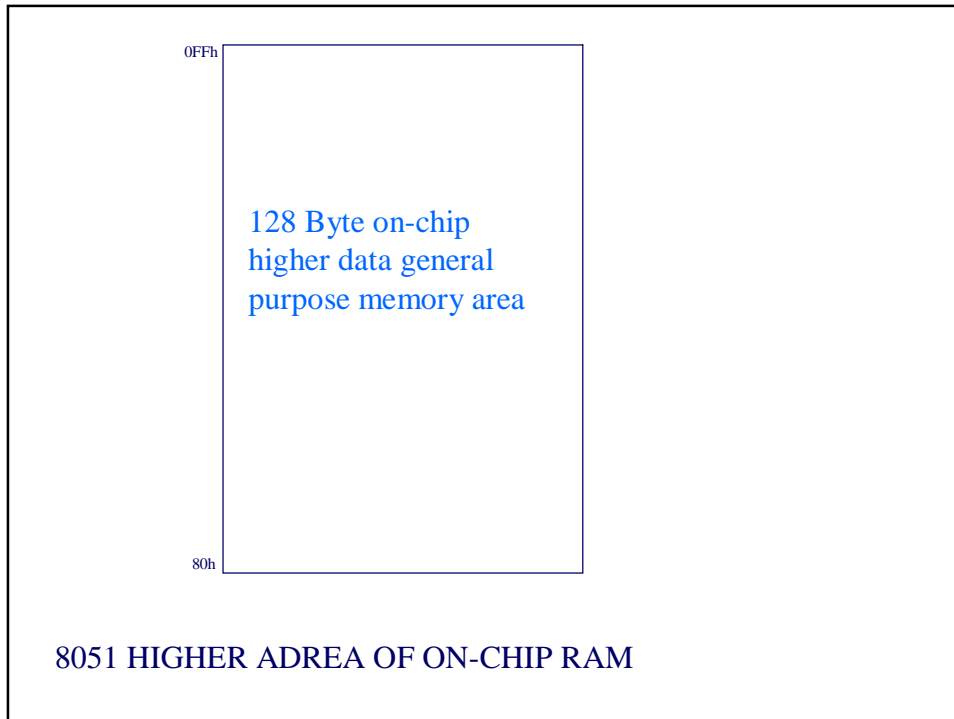


STRUCTURE OF 8051









240 (0F0H)	<b>B</b>	<b>general purpose register</b>
224 (0E0H)	<b>ACC</b>	<b>accumulator</b>
208 (0D0H)	<b>PSW</b>	<b>program status word</b>
184 (0B8H)	<b>P</b>	interrupt priority register
176 (0B0H)	P 3	i/o port 3
168 (0A8H)	<b>IE</b>	interrupt mask
160 (0A0H)	P 2	i/o port 2
153 (99H)	<b>SBUF</b>	<b>serial port data register</b>
152 (98H)	<b>SCON</b>	<b>serial port control register</b>
144 (90H)	P 1	i/o port 1
141 (8DH)	<b>TH1</b>	timer 0 data register higher part
140 (8CH)	<b>TH0</b>	timer 0 data register lower part
139 (8BH)	<b>TL1</b>	timer 1 data register higher part
138 (8AH)	<b>TL0</b>	timer 1 data register lower part
137 (89H)	<b>TMOD</b>	timer 0 and 1 mode register
136 (88H)	<b>TCON</b>	timer 0 and 1 control register
135 (87H)	<b>PCON</b>	Rejestr sterujący stanami uśpienia
131 (83H)	<b>DPH</b>	<b>DPTR register higher part</b>
130 (82H)	<b>DPL</b>	<b>DPTR register lower part</b>
129 (81H)	<b>SP</b>	<b>stack pointer</b>
128 (80H)	P 0	i/o port 0

**8051 SPECIAL FUNCTION REGISTERS**

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
C	AC	F0	RS1	RS0	OV	-	P

C	Carry Flag
AC	Auxiliary Carry Flag
F0	not used
RS1	Register bank select higher bit
RS0	Register bank select lower bit
OV	overflow
F1	not used
P	P - Parity bit

RS1	RS0	
0	0	bank 0 selected
0	1	bank 1 selected
1	0	bank 2 selected
1	1	bank 3 selected

## 8051 PROGRAM STATUS WORD REGISTER (PSW)

Tryb adresowania	Sposób wyznaczania adresu	Zasoby adresowane w tym trybie
REJESTROWY	Operandem jest rejestr R0-R7, A, B, DPTR(16bitowy) lub C(1bitowy)	Rejestry R0-R7 z aktywnego banku, A, B, DPTR(16bitowy) lub C(1bitowy)
BEZPOŚREDNI	Adres operandu jest podany explicite w rozkazie	Dolny obszar pamięci wewnętrznej RAM, rejestry SFR, bity adresowalne bezpośrednio
NATYCHMIASTOWY	Operandem jest stała znajdująca się w pamięci programu. Operand jest jednym z bajtów rozkazu. Adres wyznacza rejestr PC	Pamięć programu
POŚREDNI ZAWARTOŚCIĄ REJESTRU	Adres operandu wyznacza zawartość rejestru R0, R1, DPTR lub (dla rozkazów PUSH i POP) SP	Wewnętrzna i zewnętrzna pamięć danych RAM
POŚREDNI SUMĄ ZAWARTOŚCI REJESTRU BAZOWEGO I INDEKSU	Adres operandu wyznacza sumę zawartości rejestrów DPTR i A lub PC i A	Pamięć programu

## 8051 TRYBY ADRESACJI

Adressing mode	Address calculation method	Resources addressed with this method
REGISTER	Register is pecified in instruction	R0-R7 regidters from selected bank, A, B, DPTR(16bit) or C(1bit) registers
DIRECT	address of memory location containing data is specified in instruction	On-chipRAM lower area, SFR registers, bits from bit adressable area
IMMEDIATE	Operand is located as a part of instruction.	Program memory
REGISTER INDIRECT	Operand address is containd in adress register R0, R1, DPTR or SP	Internal and external data RAM
INDIRECT INDEKSED	Operand address is a sum of DPTR and A registers or PC and A registers content	Program memory

## 8051 ADDRESSING MODES

- **Assembler INSTRUCTIONZapis rozkazu w**

*<INSTRUCTION CODE> <argument1>,<argument 2>*

examples:

```

ADD    A,R0
MOV    A,@R1
MOV    R0,#40H
DA
CPL    A

```

## 8051 INSTRUCTION SET

- **arguments in 8051 Assembler**
  - 12345 - decimal number
  - 12345H – hexadecimal number
  - 10110000101B – binary number
  - A - accumulator
  - B - B register
  - Rn - R0...7 register
  - direct - register or memory location address (decimal hexadecimal or binary number, for example in instruction MOV A,166)
  - bit - bit address
  - @Ri - indirect addressing mode with R0 or R1 register
  - @DPTR - indirect addressing mode with DPTR register
  - #data - 8bit constant (immediate mode), e.g. #04FH
  - #data16 - 16bit constant (immediate mode), e.g. #0F06H
  - addr16 - 16 bit jump address (LJMP instruction) or subroutine address (LCALL)
  - addr11 - 11 bit jump address (AJMP) or subroutine address (ACALL)
  - rel - integer with sign (U2 code) determining short jump SJMP or conditional jump distance (short forward or backward jump with relation to actual PC content)

## 8051 INSTRUCTION SET

- **Instruction examples rozkazów**
  - ◆ **ADD A,<source>**
    - example:
    - > **ADD A,R0** function A:= (A)+(R0)
    - > **ADD A,8** A:= (A)+(8)
    - > **ADD A,#8** function A:= (A)+8
  - ◆ **MOV <destination>,< source >**
    - example :
    - MOV A, R7 ;mov to A content of R7 register
    - MOV R7, A ; mov to R7 content of A register
    - MOV R0,#30H ; mov constant 30H to R0 register
    - MOV R0, 30H ; mov content of 48th byte of on-chip RAMu to R0 register
    - MOV 30H, R0 ; in the opposite direction;
    - MOV 30H, #45H ; mov to 48(30H)th byte of RAM constant 45H
    - MOV A, @R0 ; mov to accumulator content of
    - MOV @R0, A ;in the opposite direction
    - MOV @R0, 30H ; mov to RAM loction pointed by a content of R0 register content of 30Hth byte of RAM
    - MOV @R0, #30H ; mov to RAM location pointed by content of R0 register constant 30H
    - MOV A, #35H ; load to accumulator constant 35H
    - MOV A, 35H ; load to accumulator content of 35Hth byte of on-chip RAM
    - MOV 35H, A ; mov in the opposite dirrection
    - MOV 35H, 30H ; mov the 30Hth location of RAM to
    - MOV P2, P1 ; mov the content of SFR Port 1 do SFR Port 2
    - MOV 35H, @R1 ; mov to 35Hth byte of RAM content of RAM location pointed by R1 register
  - ◆ **MOV DPTR,#data16**
    - example :
    - MOV DPTR,#01FCH ;load DPTR:=01FCH

## 8051 INSTRUCTION SET EXAMPLES