AVR CPU Registers

Microcontrollers and AVR Specific Information

Credit to Dr. Robucci for slide information

CPU Registers

Figure 2-1. Block Diagram
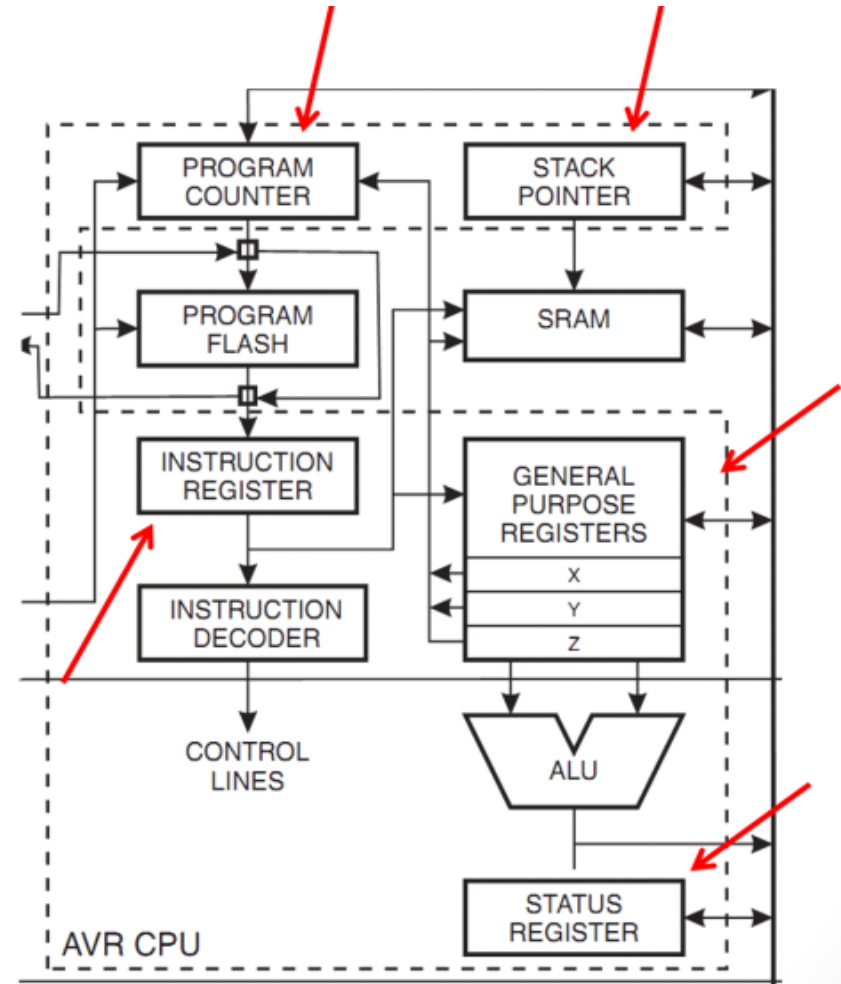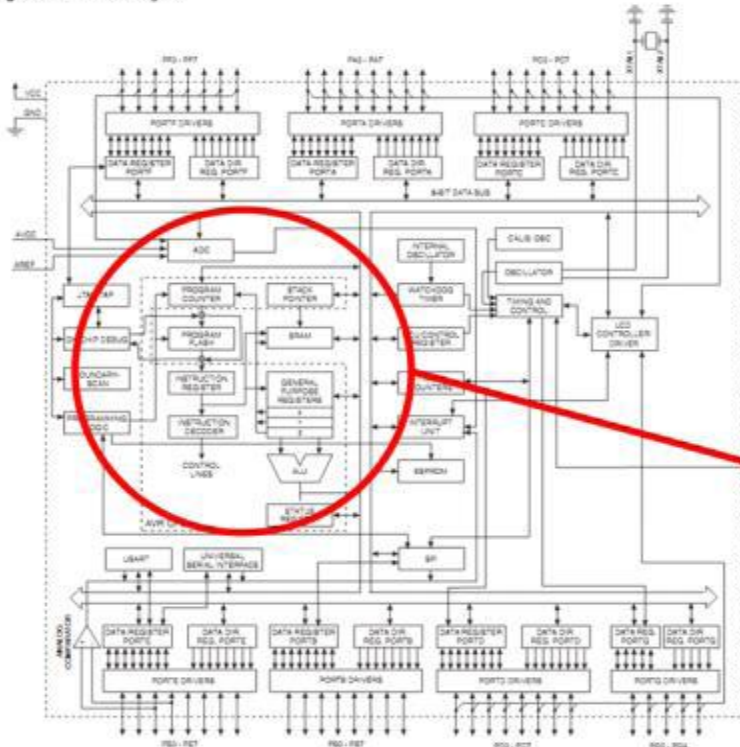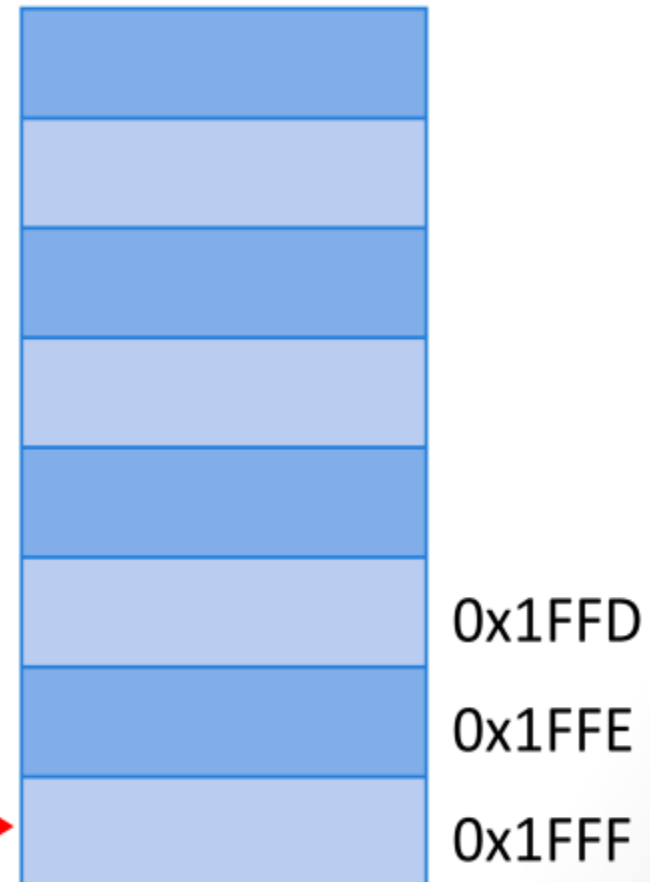
1: General Purpose Registers
2: Status Register – contains info about result of last executed instruction
3: Program Counter – Address of next instruction

4: Instruction Register – Holds fetched instruction
5: Stack Pointer – Points to top of stack

Stack

- Used for storing temporary data
  - Local variables
  - Return addresses after interrupts or subroutine
- Implemented as growing from higher to lower address
  - Initial pointer set equal to last address of SRAM
- Push – decreases SP
- Pop – increases SP

SP

0x1FFD

0x1FFE

0x1FFF

Stack Pointer - SP

- Points to top of the stack
  - Implemented as two 8-bit registers
- What does address in SP tell us?

SP implementation in AVR

Bit	15	14	13	12	11	10	9	8	
0x3E (0x5E)	–	–	–	–	–	SP10	SP9	SP8	SPH
0x3D (0x5D)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Status Register (SREG)

- Contains information about the result of the most recently executed instruction
  - Use to alter program flow on conditional operations
- NOT automatically stored when entering an interrupt routine
  - Must be handled by software

Status Register (SREG)

Bit	7	6	5	4	3	2	1	0	
0x3F (0x5F)	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

I: Global Interrupt Enable – Allows all interrupts

T: Bit Copy Storage – Used with BLD and BST for loading and storing bits from one register to another

H: Half Carry Flag – Indicates half-carry in some arithmetic

S: Sign flag – always XOR of V and N

V: Two's complement overflow flag

N: Negative Flag

Z: Zero Flag

C: Carry Flag – Indicates carry in arithmetic operation

Register File

- Most instructions have direct access to all registers
  - LD, MUL
- Some only operate on R16-R31
  - ANDI, CPI, SUBI, MULS
- A few operate only on R16:R23
  - Special multiply operations
- Double word operations operate on register pairs (R24-31)
- Most single-byte register or register+immediate operations are single cycle instructions

General	7	0	Addr.	
Purpose	R0		0x00	
Working	R1		0x01	
Registers	R2		0x02	
	...			
	R13		0x0D	
	R14		0x0E	
	R15		0x0F	
	R16		0x10	
	R17		0x11	
	...			
	R26		0x1A	X-register Low Byte
	R27		0x1B	X-register High Byte
	R28		0x1C	Y-register Low Byte
	R29		0x1D	Y-register High Byte
	R30		0x1E	Z-register Low Byte
	R31		0x1F	Z-register High Byte

Special Purpose Registers – X, Y, Z

- Registers R26-R31 have a special purpose
  - In addition to allowing general purpose usage
- 16-bit address pointers for addressing data space
- Used in functions for different addressing modes
  - Fixed displacement
  - Automatic increment/decrement

Figure 6-5. The X-, Y-, and Z-registers

	15	XH		XL	0
X-register	7		0	7	0
	R27 (0x1B)			R26 (0x1A)	

	15	YH		YL	0
Y-register	7		0	7	0
	R29 (0x1D)			R28 (0x1C)	

	15	ZH		ZL	0
Z-register	7	0		7	0
	R31 (0x1F)			R30 (0x1E)	