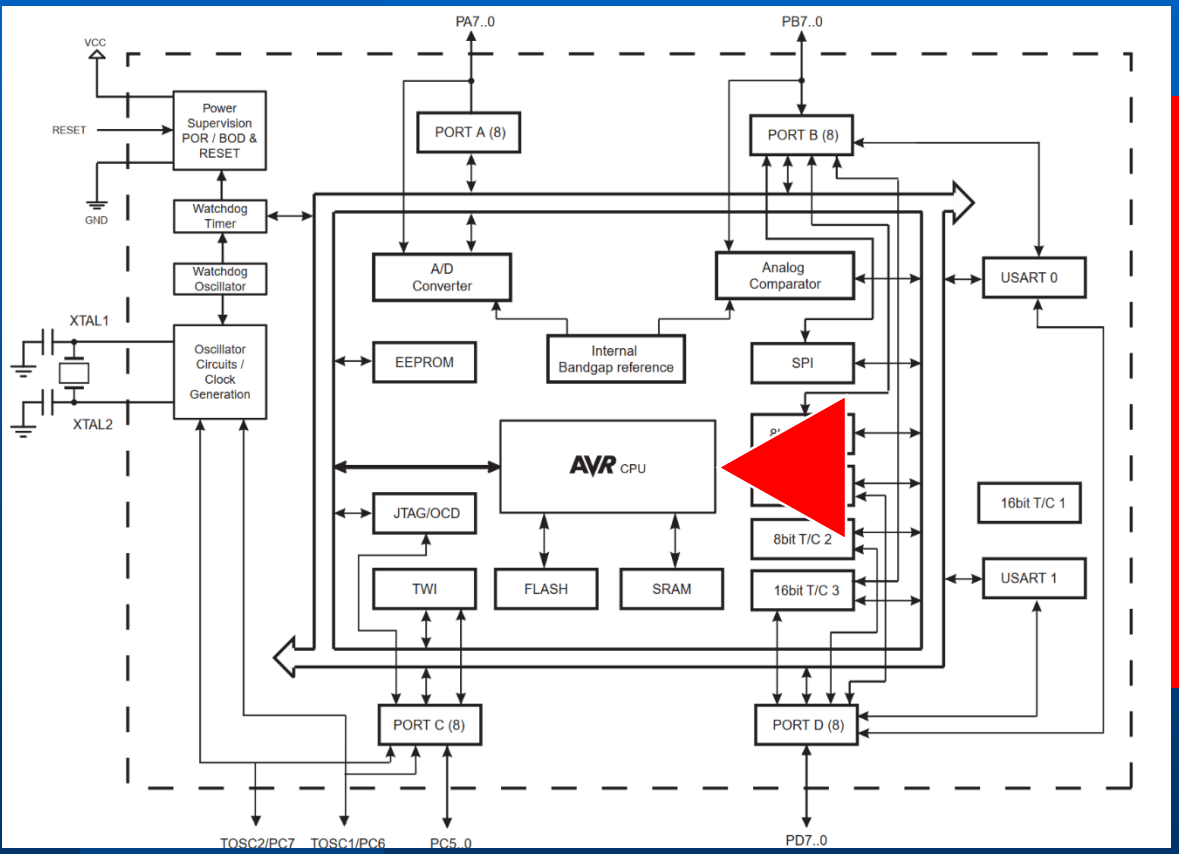
ATmega1284P Assembly II

Outline:

- The ATmega1284P Status Register
- Branch instructions
- Subroutines
- Elementary example program

The ATmega1284P Status register



SREG monitors the ALU activity

SREG is used by branch, compare and arithmetic instructions

Status Register bit definitions:

D7	D6	D5	D4	D3	D2	D1	D0
I	T	H	S	V	N	Z	C
♠	♠	♠	S=N⊕V	2's Complement Overflow	NEG.	ZERO	CARRY

- **SREG simply just another Port in the I/O region and can be accessed via IN/OUT commands**

♠ The higher bits are subject of more advanced discussions reserved for later.

ATmega1284P Datasheet Page 9:

Please look at the ATmega1284P Datasheet to get information about the Status Register (SREG) of the Microprocessor.

5.3.1 SREG – Status Register

The AVR Status Register – SREG – is defined as:

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	

- **Bit 7 – I: Global Interrupt Enable**

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

- **Bit 6 – T: Bit Copy Storage**

The Bit Copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T-bit as source or destination for the operated bit. A bit from a register in the Register File can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the Register File by the BLD instruction.

- **Bit 5 – H: Half Carry Flag**

The Half Carry Flag H indicates a Half Carry in some arithmetic operations. Half Carry Is useful in BCD arithmetic. See the “Instruction Set Description” for detailed information.

- **Bit 4 – S: Sign Bit, S = N ⊕ V**

The S-bit is always an exclusive or between the Negative Flag N and the Two’s Complement Overflow Flag V. See the “Instruction Set Description” for detailed information.

- **Bit 3 – V: Two’s Complement Overflow Flag**

The Two’s Complement Overflow Flag V supports two’s complement arithmetics. See the “Instruction Set Description” for detailed information.

- **Bit 2 – N: Negative Flag**

The Negative Flag N indicates a negative result in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

- **Bit 1 – Z: Zero Flag**

The Zero Flag Z indicates a zero result in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

- **Bit 0 – C: Carry Flag**

The Carry Flag C indicates a carry in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

Exercising the Status register

- Write a little program to read the Status Register:

```
Main:
        LDI  r16, $55
        CLR  r17
        CPI  r16, $54
        IN  r17, SREG
        OUT PORTB, r17
        rjmp Main
```

Change this number and see how the SREG bits change

CPI Rd, K : Rd – K
Flags: Z,N,V,C,H

Exercising the Status register

Exercise: Try to set the various status register bits using commands (like CPI) that would set them.

Hint: Check the documentation of the ATmega1284P assembly commands (in the course web page) and find commands that can set the appropriate flags.

Subroutines

- **Separate tasks using subroutines:**

Within your main program you can call subroutines by: ***rcall delay0***

```
Delay0:
        INC  r16
        BREQ  finito
        rjmp Delay0
finito: ret
```

Branch Instructions

- **In the previous example:**

```
Delay0:
        INC  r16
        BREQ  finito
        rjmp Delay0
finito: ret
```

The command ***BREQ*** will inspect the SREG
After the ***INC*** command has been executed and if it is found that the ZERO bit is set it will instruct the processor to jump to '***finito***'

Exercises:

Exercises during the Lab hours

Exercise 1 : Last time you made a counter. Use now the a subroutine to delay the speed of the counter.

Exercise 2 : You can delay further by cascading delay subroutines.

Homework

Homework 1: As seen in the ATMega1284P data sheet summary, each instruction requires a certain number of clock counts. The cock frequency can be found in the XPLAINED development board information documents. Using these, design a counter which counts every 1 second and sends the result to the LEDs in PORTB.