

Linnaeus University

Faculty of Technology – Department of Computer Science

1DT301 - Computer Technology Lab 2

Student: <Mohammadali> <Rashidfarokhi>

Student ID: mr223jp@student.lnu.se



COMPUTER TECHNOLOGY LAB 02

1. Explain the process of triggering an interrupt in AVR.

At the point when an interrupt happens, the current program execution is halted, the setting is spared and the control bounces to Interrupt Service Routine (the ISR). At the point when the ISR is executed, the principle program execution has proceeded. This is typically the situation with the most straightforward Micro-regulators. We have been working with Atmega32 for this arrangement, anyway, there should not be a lot of distinction making it work with other AVR MCUs. Presently, let us see how to design the External Interrupts in AVR.

Steps to arrange the Interrupts:

- Set INT1 and INT0 bits in the General Interrupt Control Register (GICR)
- Arrange MCU Control Register (MCUCR) to choose interfere with type.
- Set Global Interrupt(I-bit) Enable piece in the AVR Status Register(SREG)
- Handle the hinder in the Interrupt Service Routine code.

INT1 and INT0 empower the two interrupts. MCUCR helps in arranging the kind of intrude on, level, edge set off, and so on. The I-bit in SREG is the ace control for all interrupts in AVR miniature regulator. Watch the grouping it is turned on after all the interrupts are arranged. This forestalls, any hinder to happen before the rest of them are designed.

2. What an Opcode and Identify the opcode for the assembler instructions ADD and SUB.

An opcode is a solitary guidance that can be executed by the CPU. ... The opcode is the MOV guidance. Different parts are known as the 'operands'. Operands are controlled by the opcode. In this model, the operands are the register named AL and the worth 34 hex.

ADD → 0000 11rd

SUB → 0001 10rd

3. Difference Between Assembly and Machine Code.

• Machine Code

A developer composes PC programs utilizing significant level programming dialects. These dialects have a basic and effectively reasonable linguistic structure, like the English language. C, C++, Python, Java are a few instances of significant-level programming dialects. Nonetheless, the CPU does not comprehend these projects or source codes. Subsequently, it is important to change over these elevated level projects into machine-justifiable machine code. The compiler or a mediator plays out this transformation.

• Assembly Language

Assembly language is a middle of the road language between elevated level language and machine code. It is one level above machine code and one level beneath elevated level languages. Also, it has a punctuation like English, however,

it is more troublesome than elevated level programming languages. Assembly language is nearer to the equipment level. Hence, it is viewed as a low-level language. In this, the software engineer ought to have a decent comprehension of the PC design and register structure to compose programs in the Assembly. At that point, a constructing agent changes over the assembly language program into machine code. Subsequently, this language is more helpful for building ongoing, installed frameworks.

4. Stack in AVR Microcontrollers, why is it useful

➤ Stacks

- FIFO and LIFO
- SP
- Initialization

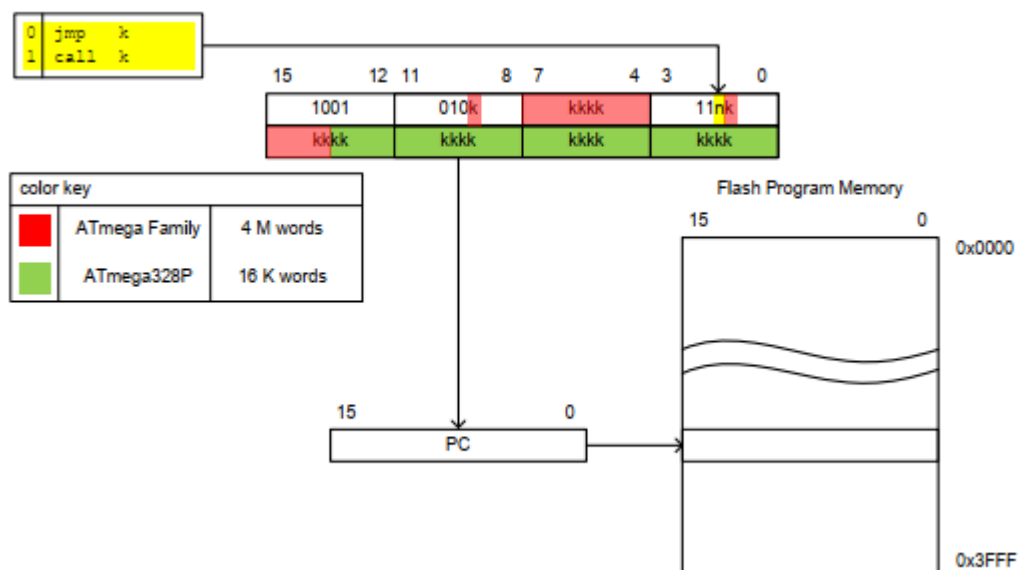
➤ LIFO Stack Operations (Push and Pop)

- Explicit push and pop

The Program Counter byte ordering on the SRAM stack is Big Endian.

All control transfer addressing modes modify the program counter.

The Program Counter byte ordering on the SRAM stack is Big Endian.



Bibliography

- [1] https://exploreembedded.com/wiki/AVR_External_Interrupts
- [2] Stopforth, Riaan. (2020). Points to take note for the Programming of Microcontrollers - focusing on ATMEL ATMEGA 328P and CodeVisionAVR. 10.13140/RG.2.2.23829.68323.
- [3] The AVR Microcontroller and Embedded Systems using Assembly and C by Muhammad Ali Mazidi, Sarmad Naimi, and Sepehr Naimi - Section: 3.2