Microprocessor Features

- **16bit Modified Harvard Arhitecture**
  - **64k program memory address space**
  - **32bit program memory organization**
  - **64k data memory address space**
  - **16bit data memory organization**

- **5-Stage Pipeline**
  - **1st stage: Instruction Fetch and IRQ Control**
  - **2nd stage: Decode, Jump Prediction and Call Stack Control**
  - **3rd stage: Operand Read, Data and Flags Forwarding**
  - **4th stage: Instruction Execution**
  - **5th stage: Data Write Back, Port Operation**

- **RISC, Load-Store Instruction Set**
  - **28 Arithmetic Instructions**
  - **14 Data Transfer Instructions**
  - **15 Branch Instructions**
  - **3 CPU Control Instructions**
  - **Most instructions are executed in a single clock cycle**
  - **Branch and Call – Ret Instructions require 2 clock cycles**
  - **MULT instruction execution requires 10 clock cycles**
  - **DIV instruction execution requires 19 clock cycles**

- **16 General Purpose Registers**
  - **Registers are named R0 through R15**
  - **All registers are 16bit wide**
  - **If Masking Flag (M) set, R15 is the IRQ Mask**
  - **All Registers are set to zero after Reset**

- **8 Status and Control Flags**
  - **Masking Flag (M) – If set, enables IRQ Masking**
  - **Enable Flag (E) – If set, the overflow condition of a signed arithmetic instruction generates the Signed Overflow Interrupt**
  - **Interrupt Flag (I) – If set, enables the execution of interrupts**
  - **Signed Flag (S) – If set, all arithmetic instructions treat the operands and result as signed numbers. Also, if set, SHR behaves as an arithmetic shift right instruction**
  - **Carry Flag (C) – Depeding on the instruction, acts as Carry In, Carry Out, Borrow In or Borrow Out**
  - **Overflow Flag (O) – Overflow condition of a signed arithmetic instruction**
  - **Negative Flag (N) – Set if the last signed arithmetic instruction result was negative**
  - **Zero Flag (Z) – Set if the last arithmetic instruction result was null**

- **Powerful SET Flags instruction can set or clear multiple flags in a single clock cycle**
- **All Flags are set to zero after Reset**

➢ **4 Input Ports**
- **Input ports are named PINA through PIND**
- **All input ports are 16bit wide**

➢ **4 Output Ports**
- **Output ports are named PORTA through PORTD**
- **All output ports are 16bit wide**
- **Powerful OUT instruction can use the current value of an output port to calculate the next value of the port (for bit manipulation)**
- **All Ports are set to zero after Reset**

➢ **Interrupt Sistem**
- **4 internal and 12 external interrupts**
- **Internal interrupts**
  - **DIVISION BY ZERO – Level 0 (Highest) Priority – Handler at address: 0x0001**
  - **CALL STACK OVERFLOW – Level 1 Priority – Handler at address: 0x0002**
  - **DATA STACK OVERFLOW – Level 2 Priority – Handler at address: 0x0003**
  - **SIGNED OVERFLOW – Level 3 Priority – Handler at address: 0x0004**
- **External interrupts from IRQ_EXT pins**
  - **IRQ_EXT [0] – Level 4 Priority – Handler at address: 0x0005**
  - **…**
  - **IRQ_EXT [11] – Level 15 Priority – Handler at address: 0x0010**
- **All interrupt requests are buffered and executed by priority level**
- **The external interrupt pins have to be held high for at least one clock cycle to be buffered. It does not matter how long the pin is held high, the interrupt will be executed just one time**
- **If Interrupt Flag (I) is not set, all interrupt requests are ignored**
- **if Enable Flag (E) is not set, the SIGNED OVERFLOW interrupt request is ignored**
- **if Masking Flag (M) is set, R15 acts as an IRQ Mask for both internal and external interrupt requests. For example, if M = 1 and R15 = 0xFFFB, the DATA STACK OVERFLOW interrupt request is ignored**
- **Interrupts are nested by default, but the Interrupt Flag can be cleared manually in the interrupt handler**
- **If the CPU is in STOP condition, the interrupt sistem is disabled**

➢ **Hardware Implemented Stacks**
- **1024 x 16b CALL STACK for Program Counter storage**
- **4096 x 16b DATA STACK for Registers and Flags storage**
- **Generate CALL STACK OVERFLOW and DATA STACK OVERFLOW Interrupt Requests**

- **Serial Multiplier**
  - **Booth Recoding Radix-4 Algorithm**
  - **16 bit Inputs, 16 bit Output**
  - **Execution in 10 clock cycles**
  - **Signed or Unsigned multiplication depending on the Signed Flag**
  - **Sets the Carry Flag if the result is wider than 16 bit**
  - **First operand from Register, second operand from Register (MULT) or Immediate Data (MULTI)**

- **Serial Divider**
  - **Streamlined Arhitecture**
  - **16 bit Inputs, 16 bit Quotient Output, 16 bit Remainder Output**
  - **Execution in 19 clock cycles**
  - **Signed or Unsigned division depending on the Signed Flag**
  - **Sets the Carry Flag if Remainder is not null**
  - **First operand (dividend) from Register, second operand (divisor) from Register (DIV) or Immediate Data (DIVI)**
  - **Stores the Quotient in the destination Register (Rd)**
  - **Stores the Remainder in the Rd + 1 Register. If Rd = R15, the Remainder will be stored in R0 Register**
  - **Generates DIVISION BY ZERO Interrupt Request if the divisor is null**