RISC V 32 Integer Base Processor

Overall Design

Address Mapping

|  |  |  |  |
| --- | --- | --- | --- |
| **Starting Address** | **Ending Address** | **Size** | **Description** |
| 0x0000\_0000 | 0x000F\_FFFF | 1MB | Boot RAM |
| 0x0010\_0000 | 0x001F\_FFFF | 1MB | Boot ROM |
|  |  |  | Reserved |
| Peripherals | | | |
| SPI | | | |
|  |  | 16KB | SPI0 |
|  |  | 16KB | SPI1 |
|  |  |  | Reserved |
| UART | | | |
|  |  | 16KB | UART0 |
|  |  |  | Reserved |
| GPIO | | | |
|  |  | 16KB | GPIO0 |
|  |  |  | Reserved |
| Memory | | | |
| 0x6000\_0000 | 0xDFFF\_FFFF | 2048MB | DDR |
| 0xEFFF\_FFFF | 0xFFFF\_FFFF |  | Reserved |

Memory Scheme

* SPI Flash
* SD
* DDR

Peripherals

* UART: For serial communication
* SPI: 2 SPI for SD and SPI flash
* GPIO: General Purpose IO

Architecture

1. Boot Up Controller

The instructions from SPI flash (BOOT ROM) are copied to BOOT RAM. After transfer the reset to RISC V core is de-asserted.

* Transfer size : 2KB

1. Boot RAM Interface

Dual Port BRAM is used as the primary boot ram for the processor. While data is copied from BOOT ROM to BOOT RAM, it runs at 75MHz. After transfer during the instruction access and data access it runs at 100MHz.

* Reference freq : 75MHz during boot up & 100MHz after boot
* Width :
* Depth :
* Total Size :
* IP : Block Memory Generator v7.3

1. Boot ROM

SPI flash is used as the BOOT ROM.

1. RV32I Core
2. UART Controller
3. GPIO Controller
4. SPI Controller

**RV32I Core**

Basic Details:

|  |  |
| --- | --- |
| Core Specification | |
| Architecture | 32-bit |
| Pipeline Stages | 6 stages |
| Privilege Leve | M |
| Instruction | Base Integer(I) |
| Frequency | 100 MHz (Spartan 6) |
| Endian | Little Endian |

PIPELINE

1. Instruction Fetch Unit

Fetches the instruction based on the PC (Program Counter). After each cycle the PC is incremented with specific value (4 if 8bit store mode and 1 if 32 bit mode).

|  |  |  |
| --- | --- | --- |
| Port Name | Width | Description |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

A control Unit present in the Fetch unit to control instruction read based on input from halt control. The details about the halt controller is explained in the corresponding section.

1. Instruction Decode Unit

Decodes the instruction as per RISC V specification and forwards the decoded instruction to Execution Unit.

There is a data forwarding path if execution data address and fetch address are same then the data from execution is forwarded from execution unit

1. Execution Unit

Executes the data as per decoded instruction.

1. Memory Access
2. Register Write Back
3. Halt Controller
4. Interrupt Controller

CSR Registers

1. Machine ISA Register(misa)

* CSR Address : 0x301
* Description : ISA and extensions
* Default Value : 0x4000\_0100
* R/W Access : MRW

|  |  |  |  |
| --- | --- | --- | --- |
| Bit Field | 31-30 | 29-26 | 25-0 |
| Purpose | MXL | 0 | Extension |
| Number of bits | 2 | 4 | 26 |
| RD/WR | RD/WR | RO | RD/WR |
| Default Bit Value | 0x1 | 0x0 | 0x100 |

*Notes:*

* *Extension Supported: I (Base on Table 3.2 of RISC V Privileged Spec)*
* *MXL: 1 (Base on Table 3.1 of RISC V Privileged Spec)*

1. Machine Vendor ID Register (mvendorid)

* CSR Address : 0xF11
* Description : Vendor ID
* Default Value : 0x0000\_0000
* R/W Access : MRO

|  |  |  |
| --- | --- | --- |
| Bit Field | 31-7 | 6-0 |
| Purpose | Bank | Offset |
| Number of bits | 25 | 7 |
| RD/WR | RO | RO |
| Default Bit Value | 0x0 | 0x0 |

*Notes*

* *Value can be 0 showing not implemented/non-commercial implementation*

1. Machine Architecture ID Register (marchid)

* CSR Address : 0xF12
* Description : Architecture ID
* Default Value : 0x0000\_0000
* R/W Access : MRO

|  |  |
| --- | --- |
| Bit Field | 31-0 |
| Purpose | Architecture ID |
| Number of bits | 32 |
| RD/WR | RO |
| Default Bit Value | 0x0 |

*Notes:*

* *Value can be 0 showing not implemented/non-commercial implementation*

1. Machine Implementation ID Register (mimpid)

* CSR Address : 0xF13
* Description : Architecture ID
* Default Value : 0x0000\_0000
* R/W Access : MRO

|  |  |
| --- | --- |
| Bit Field | 31-0 |
| Purpose | Implementation |
| Number of bits | 32 |
| RD/WR | RO |
| Default Bit Value | 0x0 |

*Notes:*

* *Value can be 0 showing not implemented*

1. Hart ID Register (mhartid)

* CSR Address : 0xF14
* Description : Hardware thread ID
* Default Value : 0x0000\_0000
* R/W Access : MRO

|  |  |
| --- | --- |
| Bit Field | 31-0 |
| Purpose | Hart ID |
| Number of bits | 32 |
| RD/WR | RO |
| Default Bit Value | 0x0 |

*Notes:*

* *Value can be 0 showing not implemented*

1. Machine Status Register (mstatus and mstatush)

* CSR Address : 0x300 & 0x310
* Description : Pointer to configuration data structure
* Default Value : 0x0000\_1800 & 0x0000\_0000
* R/W Access : MRW

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bit Field | 31 | 30-23 | 22 | 21 | 20 | 19 | 18 | 17 | 16-15 | 14-13 |
| Purpose | SD | WPRI | TSR | TW | TVM | MXR | SUM | MPRV | XS | FS |
| Number of bits | 1 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 |
| RD/WR | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO |
| Default Bit Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bit Field | 12-11 | 10-9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Purpose | MPP | VS | SPP | MPIE | UBE | SPIE | WPRI | MIE | WPRI | SIE | WPRI |
| Number of bits | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| RD/WR | RO | RO | RO | RW | RO | RO | RO | RW | RO | RO | RO |
| Default Bit Value | 0x3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table: mstatus register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bit Field | 31-6 | 5 | 4 | 3-0 |
| Purpose | WPRI | MBE | SBE | WPRI |
| Number of bits | 26 | 1 | 1 | 4 |
| RD/WR | RO | RD/WR | RO | RO |
| Default Bit Value | 0 | 0 | 0 | 0 |

Table: mstatush register

*Notes:*

* *MIE: Global Interrupt Enable for machine mode*
* *MPIE: Holds the value of interrupt-enable bit active prior to trap*
* *MPP: Holds the previous privilege mode (2’b11 Machine Mode)*
* *MPRV: Read only 0 if U-Mode not supported*
* *MXR: Read only 0 if S-Mode not supported*
* *MBE: Controls whether non-instruction-fetch memory access is Little (0)/Big endian (1)*

1. Machine Trap-Vector Base Address Register (mtvec)

* CSR Address : 0x305
* Description : Machine trap handler base address
* Default Value : 0x0000\_0000
* R/W Access : MRW

|  |  |  |
| --- | --- | --- |
| Bit Field | 31-2 | 1-0 |
| Purpose | Base | Mode |
| Number of bits | 30 | 2 |
| RD/WR | RW | RO |
| Default Bit Value | 0x0 | 0x0 |

*Notes:*

* *Mode: Direct (Based on Table 3.5 of Privilege Spec)*
* *Base: address need to be 4 bytes aligned.*

1. Machine Trap Delegation Address Register (medeleg & mideleg)

* CSR Address : 0x302 & 0x303
* Description : Machine exception and interrupt delegation register
* Default Value : 0x0000\_0000
* R/W Access : MRW

|  |  |
| --- | --- |
| Bit Field | 31-0 |
| Purpose | Exception Delegation |
| Number of bits | 32 |
| RD/WR | RO |
| Default Bit Value | 0x0 |

Table: medeleg register

|  |  |
| --- | --- |
| Bit Field | 31-0 |
| Purpose | Interrupt Delegation |
| Number of bits | 32 |
| RD/WR | RO |
| Default Bit Value | 0x0 |

Table: mideleg register

*Notes:*

* *If S-mode not implemented medeleg and mideleg should not exist*

1. Machine Interrupt Register (mip & mie)

* CSR Address : 0x344 & 0x304
* Description : Machine interrupt pending and enable registers
* Default Value : 0x0000\_0000
* R/W Access : MRW

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bit Field | 31-16 | 15-12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Purpose | Custom Interrupts | 0 | MEIP | 0 | SEIP | 0 | MTIP | 0 | STIP | 0 | MSIP | 0 | SSIP | 0 |
| Number of bits | 16 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| RD/WR | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO |
| Default Bit Value | 0x0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 |

Table: mip register

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bit Field | 31-16 | 15-12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Purpose | Custom Interrupts | 0 | MEIEE | 0 | SEIE | 0 | MTIE | 0 | STIE | 0 | MSIE | 0 | SSIE | 0 |
| Number of bits | 16 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| RD/WR | RO | RO | RW | RO | RO | RO | RW | RO | RO | RO | RW | RO | RO | RO |
| Default Bit Value | 0x0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 |

Table: mie register

*Notes:*

* *MEIE: interrupt enable bit for machine level external interrupt*
* *MEIP: interrupt pending read only bit for machine level external interrupt. Set and cleared by interrupt controller*
* *MTIE: interrupt enable bit of machine timer*
* *MTIP: interrupt pending read only bit of machine timer. Set and cleared by timer compare register*
* *MSIE: interrupt enable bit for software interrupt*
* *MSIP: interrupt pending read only bit for software interrupt. Set and cleared by memory mapped control register.*

1. Hardware Performance Monitor (mcycle(h), minstret(h) & mhpmcounter3-31(h))

* CSR Address : 0xB00 – 0xB1F & 0xB80 – 0xB9F
* Description : Machine cycle & performance monitor counter
* Default Value : 0x0000\_0000
* R/W Access : MRW

|  |  |
| --- | --- |
| Bit Field | 31-0 |
| Purpose | *mcycle, minstret, mphmcountern* |
| Number of bits | 32 |
| RD/WR | RO |
| Default Bit Value | 0x0 |

Future Update:

* Implementation of all hardware monitors

*Notes:*

* *For RV32 both mcycleh, minstreth, mphmcounter3h-31h should also be implemented*
* *Value of 0 is legal*

1. Machine Counter Enable (mcounteren)

* CSR Address : 0x306
* Description : Machine counter enable
* Default Value : 0x0000\_0000
* R/W Access : MRW

|  |  |
| --- | --- |
| Bit Field | 31-0 |
| Purpose | *mcounteren* |
| Number of bits | 32 |
| RD/WR | RO |
| Default Bit Value | 0x0 |

*Notes:*

* *For implementation without U-mode mcounteren should not exist.*

1. Machine Counter-Inhibit Register (mcountinhibit)

* CSR Address : 0x320
* Description : Machine counter-inhibit
* Default Value : 0x0000\_0000
* R/W Access : MRW

|  |  |
| --- | --- |
| Bit Field | 31-0 |
| Purpose | *mcountinhibit* |
| Number of bits | 32 |
| RD/WR | RO |
| Default Bit Value | 0x0 |

*Notes:*

* *If the mcountinhibit register is not implemented, the implementation behaves as though the register were set to zero*

1. Machine Scratch Register (mscratch)

* CSR Address : 0x340
* Description : Scratch register for machine trap handler
* Default Value : 0x0000\_0000
* R/W Access : MRW

|  |  |
| --- | --- |
| Bit Field | 31-0 |
| Purpose | mscratch |
| Number of bits | 32 |
| RD/WR | RO |
| Default Bit Value | 0x0 |

1. Machine Exception Program Counter (mepc)

* CSR Address : 0x341
* Description : Machine exception program counter
* Default Value : 0x0000\_0000
* R/W Access : MRW

|  |  |  |
| --- | --- | --- |
| Bit Field | 31-2 | 1-0 |
| Purpose | *mepc* | 0 |
| Number of bits | 30 | 2 |
| RD/WR | RW | RO |
| Default Bit Value | 0x0 | 0 |

1. *Machine Cause Register (mcause)*

* *CSR Address : 0x342*
* *Description : Machine trap cause*
* *Default Value : 0x0000\_0000*
* *R/W Access : MRW*

|  |  |  |
| --- | --- | --- |
| Bit Field | 31 | 30-0 |
| Purpose | Interrupt | 0 |
| Number of bits | 1 | 2 |
| RD/WR | RW | RO |
| Default Bit Value | 0x0 | 0 |

*Notes:*

* *Interrupt and exception codes are defined in Table 3.6 of RV Privileged doc*

1. Machine Trap Value Register (mtval)

* CSR Address : 0x343
* Description : Machine bad address or instruction
* Default Value : 0x0000\_0000
* R/W Access : MRW

|  |  |
| --- | --- |
| Bit Field | 31-0 |
| Purpose | mtval |
| Number of bits | 32 |
| RD/WR | RW |
| Default Bit Value | 0 |

1. Machine Configuration Pointer Register (mconfigptr)

* CSR Address : 0xF15
* Description : Pointer to configuration data structure
* Default Value : 0x0000\_0000
* R/W Access : MRO

|  |  |
| --- | --- |
| Bit Field | 31-0 |
| Purpose | mconfigptr |
| Number of bits | 32 |
| RD/WR | RO |
| Default Bit Value | 0 |

*Notes:*

* *It may be zero to indicate the configuration data structure does not exist or that an alternative mechanism must be used to locate it.*

1. Machine Environment Configuration Register (menvcf & menvcfgh)

* CSR Address : 0xF15
* Description : Pointer to configuration data structure
* Default Value : 0x0000\_0000
* R/W Access : MRW

|  |  |
| --- | --- |
| Bit Field | 31-0 |
| Purpose | Menvcf & menvcfgh |
| Number of bits | 32 |
| RD/WR | RO |
| Default Bit Value | 0 |

*Notes:*

* *If U Mode is not supported menvcf & menvcfgh doesn’t exist*

1. Machine Timer Register (mtime & mtimecmp)

* CSR Address : 0xF15
* Description : Pointer to configuration data structure
* Default Value : 0x0000\_0000
* R/W Access : MRW

|  |  |
| --- | --- |
| Bit Field | 63-0 |
| Purpose | mtime & mtimecmp |
| Number of bits | 32 |
| RD/WR | RO |
| Default Bit Value | 0 |

*Notes:*

* *The platform must provide a mechanism for determining the period of an mtime tick.*
* *The mtime register will wrap around ifthe count overflows.*
* *Platforms provide a 64-bit memory-mapped machine-mode timer compare register (mtimecmp).*

*Notes:*

* *Reserved Writes Preserve Values, Reads Ignore Values (WPRI): Value should be preserved even if SW writes some other values. Hardwire to zero*
* *Write/Read Only Legal Values (WLRL): only write specific bit encoding i.e., SW should not write illegal values.*
* *Write Any Values, Read Legal Values (WARL): SW can write any values but while reading only legal values are obtained*

SPARTAN 6 XC6SLX9

* Bin File Size: 343KB (From document UG380 Page: 78)
* Bin file Address
* BootROM address in SPI FLASH: 0x0010\_0000 to 0x001F\_4239 (1MB)
* Reserved file

Branch Halt

When branch\_en signal asserted

* 1 clk: decode & operand fetch & exe stage halted
* 2 clk: operand fetch & exe & wr stage halted
* 3 clk: exe & wr stage halted
* 4 clk: wr stage halted

Boot Clock : 50MHz

Core Clock : 100MHz

ELF Instruction Start Loc : 0x34