**TILE\_SS\_RVC:  
The Tile building block in the SOC  
Connects to the SS\_RVC IP**

HAS: High-Level Architecture Specifications

Revision HAS 0.65  
Amichai Ben-David ([amichai.ben.david@intel.com](mailto:amichai.ben.david@intel.com))

Table of Contents

[1 General Description 3](#_Toc79052381)

[1.1 Tile Block Diagram 3](#_Toc79052382)

[2 Top Level Interface 4](#_Toc79052383)

[2.1 Top Level Interface Tables 5](#_Toc79052384)

[3 Internal interconnect 6](#_Toc79052385)

[3.1 Internal interconnect Block Diagram 6](#_Toc79052386)

[4 Interface Block Diagram 7](#_Toc79052387)

[5 TILE\_SS\_RVC\_<#> in the SOC 8](#_Toc79052388)

[5.1 SOC Block Diagram 8](#_Toc79052389)

[5.2 SOC Execution Flow: 8](#_Toc79052390)

Figures

[Figure 1 – TILE\_SS\_RVC Block Diagram 3](#_Toc79052284)

[Figure 2 - Internal Interconnect 6](#_Toc79052285)

[Figure 3 - Interface Block Diagram 7](#_Toc79052286)

[Figure 4 - SOC Block diagram 8](#_Toc79052287)

Tables

[**Table 1 - Revision History** 2](#_Toc79441991)

[**Table 2 - Related Documents** 2](#_Toc79441992)

[**Table 3 - Glossary** 2](#_Toc79441993)

[**Table 4 – SS\_RVC Parameters** 4](#_Toc79441994)

[**Table 5 - Standard Interface Protocol** 4](#_Toc79441995)

[**Table 6 – SS\_RVC general interface** 5](#_Toc79441996)

[**Table 7 - Tile <->Tile standard interface** 5](#_Toc79441997)

[**Table 8 - Tile<->SS\_RVC standard interface** 5](#_Toc79441998)

[**Table 9 - Internal Interconnect** 6](#_Toc79441999)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| R*ev. No.* | Who | Description | Rev. Date |
| 0.1 | Amichai Ben-David | Initial TILE\_SS\_RVC HAS | 28 July 2021 |
| 0.3 | Amichai Ben-David | Move the Core outside the tile Change Core interface to standard. | 5 August 2021 |
| 0.6 | Amichai Ben-David | Remove the “Load” option from core. Core can only “STORE” -> Removing the Rsp<\*>DmQ104H interface | 5 August 2021 |
| 0.65 | Amichai Ben-David | Fix typos. Remove the word | 9 August 2021 |

**Table 1 - Revision History**

Related Documents

|  |  |  |
| --- | --- | --- |
| Name | Path | Description |
| riscv\_isa\_spec.pdf | TODO | The full RISCV Unprivileged Specification file.  Including the RV32I Baseline ISA |
| SS\_RVC\_HAS.pdf | TODO | The ss\_rvc HAS. Describes the core architecture. |

**Table 2 - Related Documents**

Glossary

|  |  |
| --- | --- |
| Term | Description |
| ISA | Instruction Set Architecture. (such as X86, ARM, RISC-V etc.) |
| IO | Input & output. |
| IP | Intellectual Property. In this case, RTL building block that can be consumed. |
| HAS | High Level Architecture Specifications. (This document) |
| MAS | Micro Architecture Specifications. Document with the microarchitecture details. |
| I\_MEM | Instruction memory – where the program is loaded and ready for execution. |
| D\_MEM | Data Memory – where the LOAD & STORE instructions read/write Data. |
| Pipeline | Common Way to parallel and utilize Hardware  <https://en.wikipedia.org/wiki/Instruction_pipelining> |
| RISC | Reduce Instruction Set Computer. (Unlike CISC -Complex Instruction Set Computer) <https://en.wikipedia.org/wiki/Reduced_instruction_set_computer> |
| Thread | A "hardware thread" is a physical CPU or core that ca run a program. |
| RISC-V | A relatively new open and free ISA. (comparable to intel X86, ARM)  <https://en.wikipedia.org/wiki/RISC-V> |
| RV32I | “RISC-V 32-bit Integer” The RISC-V baseline compatible ISA (no extensions M/A/F etc.) |
| Standard interface | Functional characteristics to allow the exchange of information between two systems |
| Word | 32-bits of data - 4 Bytes. The size of an integer in RV32I ISA. |
| Hazard | Potential source of harm. in this document when reading Outdated data, or wrongly executing Instruction. |
| Strap | Tie signals to constant value (1’b0 or 1’b1) |
| MSFF | Main & Secondary Flip Flop. (AKA Master Slave Flip Flop) |
| Clock Gating | Logic that allows to condition the MSFF clock. Functionality and power reasons. |
| Polling | Actively sampling the status of an external device. |

**Table 3 - Glossary**

# General Description

This HAS describes the High-Level-Architecture of the tile\_ss\_rvc module.  
The tile\_ss\_rvc is one of the main building block in the SOC.

Each tile contains 2 modules:

1. **IO\_CTRL**Controls the Interface with the SOC.  
   In case the input request address “matches” the local core, the request will be forwarded to the core.   
   In case an input request address “misses” local core, it lets the Requests to pass through to next tile\_ss\_rvc.
2. **MEM\_WRAP**

Contains the I\_MEM & D\_MEM (Instruction Memory & Data Memory)  
Each memory block is dual access – accessible from both CORE & IO\_CTRL.

Holds the MMIO 1 bit register “PcResetQnnnH”.   
When this register is set the Cores Program Counter will point to Instruction address 0x0.

Each Tile connects with Standard interface to both a “SS\_RVC” and a neighboring “TILE”

**SS\_RVC -** The Core IP compatible with a subset of the RISCV RV32I ISA.

## Tile Block Diagram



Figure 1 – TILE\_SS\_RVC Block Diagram

# 

# Top Level Interface

**Default Parameter Values:**

|  |  |  |
| --- | --- | --- |
| Name | Default | Description |
| INT\_LEN | 32 | Integer Size – in RV32I Spec is known as INT\_LEN |
| REGION\_BIT | 10 | Indicates the Region I\_MEM vs D\_MEM Address |
| MSB\_I\_MEM | 9 | Instruction memory address MSB – Used to calculate memory size |
| MSB\_D\_MEM | 9 | Data memory address MSB – Used to calculate memory size |

**Table 4 – SS\_RVC Parameters**

**TODO – Describe DCS and its Standard Interface.**

**Standard Interface protocol**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Size | Optional | Description |
| Valid | 1 | No | Valid |
| Opcode | 2 | No | 2’b00 – RD (Read) 2’b01 - WR (Write)  2’b10 - <reserved> 2’b11 - RD\_RSP(Read Response) |
| Address | INT\_LEN | Yes | Address[31:22] – indicates the target “Tile-ID”  Address[REGION\_BIT] – indicates Region I\_MEM vs D\_MEM  Address[9:0] – Memory offset within 1KB I/D\_MEM |
| Data | INT\_LEN | Yes | In case of WR Opcode – write data to memory In case of Rd Opcode – data should not be used.  In case of Rd RSP Opcode – Response Data to SA. |

**Table 5 - Standard Interface Protocol**

## Top Level Interface Tables

**Interface signals:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Size | Direction | Description |
| ClkQH | 1 | Input | Q Clock – a single clock domain. 100Mhz - 2Ghz |
| ResetQnnnH | 1 | Input | Active High Reset |
| TileIdStrapQnnnH | 10 | Input | used to match Request Address to local Mem by strap match. |

**Table 6 – SS\_RVC general interface**

**Tile <->Tile Standard interface signals:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Size | Direction | Description |
| Request Input – Standard Interface | | | |
| ReqValidQ500H | 1 | input |  |
| ReqOpcodeQ500H | 2 | Input |  |
| ReqAddressQ500H | INT\_LEN | Input |  |
| ReqDataQ500H | INT\_LEN | Input |  |
| Response Output - Standard Interface | | | |
| RspValidQ503H | 1 | Output |  |
| RspOpcodeQ503H | 1 | Output |  |
| RspAddressQ503H | INT\_LEN | Output |  |
| RspDataQ503H | INT\_LEN | Output |  |

**Table 7 - Tile <->Tile standard interface**

**Tile <->SS\_RVC Standard interface signals:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Size | Direction | Description |
| PC Input – Standard Interface | | | |
| PcValidQ100H | 1 | input | Valid request to read the next Instruction from I\_MEM |
| PcOpcodeQ100H | 2 | Input | Must be strapped to RD Opcode |
| PcAddressQ100H | INT\_LEN | Input | The program Counter is used as the “read address pointer” |
| PcDataQ100H | INT\_LEN | Input | **----------------------------Not in use-----------------------** |
| Instruction Output - Standard Interface | | | |
| InstValidQ101H | 1 | Output | Valid Instruction from I\_MEM |
| InstOpcodeQ101H | 2 | Output | Must be strapped to RD\_RSP Opcode |
| InstAddressQ101H | INT\_LEN | Output | **----------------------------Not in use------------------------** |
| InstDataQ101H | INT\_LEN | Output | The instruction that was read from I\_MEM to be executed |
| D\_MEM Request – Standard Interface | | | |
| ReqValidDmQ103H | 1 | input | Valid memory Access from Core |
| ReqOpcodeDmQ103H | 2 | Input | WR opcode Only (STORE instruction) |
| ReqAddressDmQ103H | INT\_LEN | Input | Address is Calculated in ALU |
| ReqDataDmQ103H | INT\_LEN | Input | Data is Read from Register file |

**Table 8 - Tile<->SS\_RVC standard interface**

**Tile <->SS\_RVC Other interface signals:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Size | Direction | Description |
| PC Input – Standard Interface | | | |
| RstPcQnnnH | 1 | output | A Register overwrite to reset the PC value |

**Table 9 - Tile<->SS\_RVC Other interface**

# 

# Internal interconnect

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Size | From | To | Description |
| Top Level | | | | |
| ClkQH | 1 | TILE\_SS\_RVC | IO\_CTRL,MEM\_WRAP |  |
| ResetQnnnH | 1 | TILE\_SS\_RVC | IO\_CTRL,MEM\_WRAP |  |
| TileIdStrapQnnnH | 10 | TILE\_SS\_RVC | IO\_CTRL | Match Request address to TILE ID |
| IO\_CTRL<-> MEM\_WRAP | | | | |
| ReqValidQ501H | 1 | IO\_CTRL | MEM\_WRAP | Standard interface |
| ReqOpcodeQ501H | 2 | IO\_CTRL | MEM\_WRAP | Standard interface |
| ReqAddressQ501H | INT\_LEN | IO\_CTRL | MEM\_WRAP | Standard interface |
| ReqDataQ501H | INT\_LEN | IO\_CTRL | MEM\_WRAP | Standard interface |
| RspValidQ502H | 1 | MEM\_WRAP | IO\_CTRL | Standard interface |
| RspOpcodeQ502H | 2 | MEM\_WRAP | IO\_CTRL | Standard interface |
| RspAddressQ502H | INT\_LEN | MEM\_WRAP | IO\_CTRL | Standard interface |
| RspDataQ502H | INT\_LEN | MEM\_WRAP | IO\_CTRL | Standard interface |

**Table 9 - Internal Interconnect**

## Internal interconnect Block Diagram



Figure 2 - Internal Interconnect

# Interface Block Diagram



Figure 3 - Interface Block Diagram

# **TILE\_SS\_RVC\_<#> in the SOC**

Many Tiles can be connected to a “ring architecture”.  
Together they make a multi-core SOC.

* All the Tiles are “passive” and cannot generate ring traffic.
* The System Agent interacts with the Tiles memory by sending RD/WR requests on the ring.
* The latency is deterministic for all transaction.
  + The Tile latency of Request to Response. (Both for TILE\_ID hit & miss):
  + The System agent Latency Request to Response:

## SOC Block Diagram



Figure 4 - SOC Block diagram

## SOC Execution Flow:

1. Reset stage – Reset the Control bits & Set the MMIO PcResetQnnnH register.

When exiting the RESET – all PC are stock at reset value pointing to 0x0 I\_MEM.

1. SA (System Agent) will load the program to I\_MEM & the Data set to D\_MEM.  
   This is done by sending the program & Data to the Tiles with the correct Address offset.
2. Start the program by writing to the “PcResetQnnnH” 1’b0.  
   This will allow the PC to exit the reset and start executing the program.