HARMONICA v2.0 Design Specification

Poulami Das

ABSTRACT

To be written

1. REVISION HISTORY

Table 1: Revision History Record

Revision	Description	Made by	Date
1.0	Initial Logic entry	Poulami Das	Spr 2018

2. DETAILED DESIGN DESCRIPTION

This section comprises of a detailed description of all the major units that are contained in the datapath of our proposed design.

2.1 General Purpose Registers

The following sub sections describe the components that make up the general purpose register file. In order to easily simulation and synthesis this module, a set of compile time switches are used in addition to module parameters. The module parameters are described for each sub module as they are described in the following sections. The compile time switches used are enlisted in Table 2.

Table 2: Compile Time options

Switch Name	Description
DATA_WIDTH_IS_1	Register width is 1
DATA_WIDTH_IS_32	Register width is 32
DATA_WIDTH_IS_64	Register width is 64
NUMBER_OF_REGISTERS_IS_32	32 registers per lane
NUMBER_OF_REGISTERS_IS_64	64 registers per lane
NUM_LANES_IS_16	16 lanes (8 default)
NUMBER_OF_WARPS_IS_16	16 warps (8 default)

2.1.1 Register File

The register file is a parameterizable block of code whose parameters are listed in Table 3. It comprises of a single write port and 2 individual read ports. The reads are asynchronous. The diagrammatic representation of the register file is presented in Figure 1. The details of the input output ports for this module is given in Table 4. The register address space numbering and reset values are described in Table 5. The test cases used to test this block are enlisted in Table 6. There are 18 testcases in total.

Table 3: Parameters for Register File

Parameter Name	Supported Values	Default
LOG2_NUM_REGS	4/5/6	4
DATA_WIDTH	32/64	32
NUM_REGS	16/32/64	16

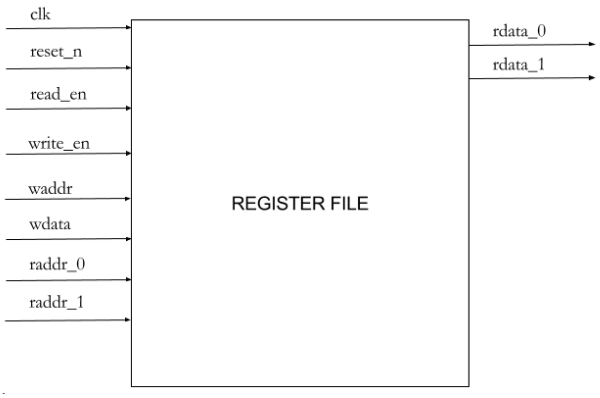

Figure 1: Block Diagram of Register File

2.1.2 Register_Bank

This module contains an instance of a register file for each lane. The number of lanes can be configured to match the number of streaming processors they connect to. The parameters available in addition the register file parameters is described in Table 7. Figure 2 shows the block diagram of a register bank. The I/O description for this module is given in Table 8. On a write, data is written to the same address of each register file. For reads, the data is read out from each register file for the same address. The test cases used to verify this system are detailed in Table 9

2.1.3 Register_Block

This module contains an instance of a register bank per warp. The number of warps by default is 8. 16 warps can be supported using compile time switch. Figure 3 shows the block diagram of a register block. The I/O description

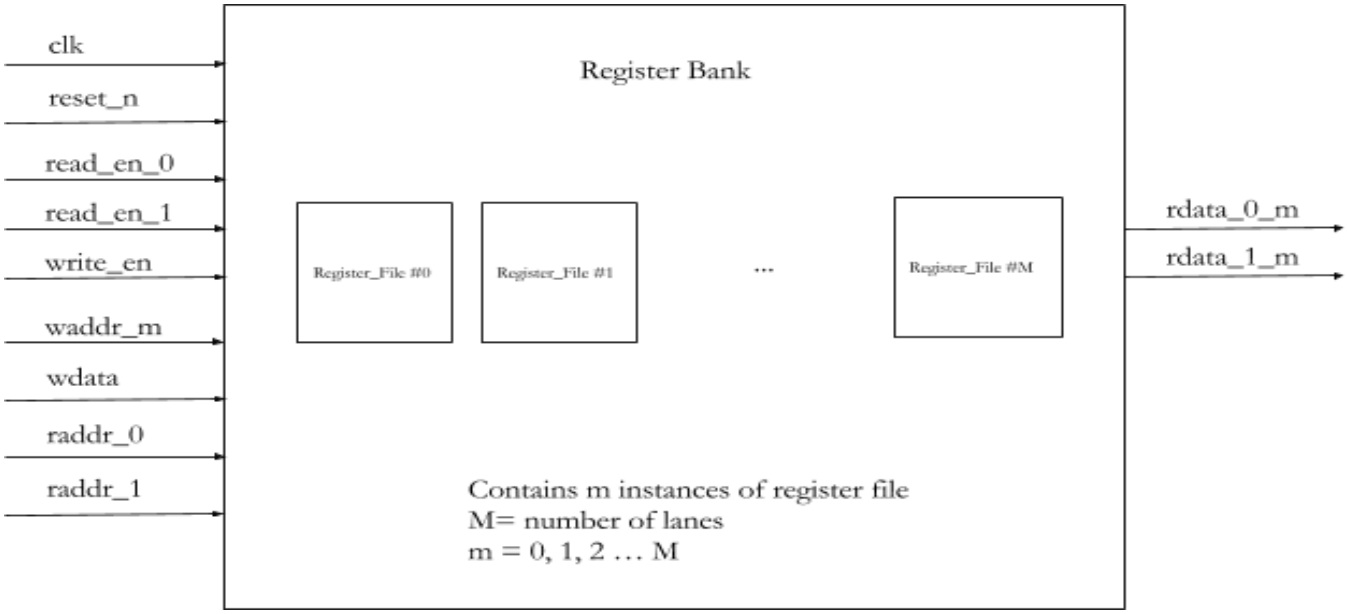

Figure 2: Block Diagram of Register Bank

Table 4: I/O Description for Register File

Name	Type	Width	Description
clk	input	1	clock source
reset_n	input	1	Active low synchronous reset
read_en	input	2	01:raddr_0 valid 10:raddr_0 and raddr_1 valid 11:raddr_0 and raddr_1 valid
write_en	input	1	waddr, wdata valid
raddr_0	input	LOG2_NUM_REGS	Read Address 0
raddr_1	input	LOG2_NUM_REGS	Read Address 1
waddr	input	LOG2_NUM_REGS	Write Address
wdata	input	DATA_WIDTH	Write Data
rdata_0	output	DATA_WIDTH	Read Data for raddr_0
rdata_1	output	DATA_WIDTH	Read Data for raddr_1

Table 5: Address Space and Reset Values for Register File

NUM_REGS	Address Space	Reset Value
16	0x0 to 0x0F	0x0
32	0x0 to 0x1F	0x0
64	0x0 to 0x3F	0x0

for this module is given in Table 10. On a write, the register bank corresponding to the warp_id is chosen and data is written to the same address of each register file in the particular register bank. For reads, the data is read out from each register file for the same address from the bank selected by the warp_id. The test cases used to verify this system are detailed in Table 11

Table 6: Register File Test Cases and Description

No.	Test Name	Description
1	default_val_test_m_n m= 16, 32, 64 n= 32, 64	Checks the default reset value
2	write_read_each_bit_m_n m= 16, 32, 64 n= 32, 64	Writes each register bit to 1, reads back using different value of read_en
3	random_write_read_m_n m= 16, 32, 64 n= 32, 6	Write random data to each register Reads back using different read_en

Table 7: Parameters for Register Bank

Parameter Name	Supported Values	Default Value
NUM_LANES	8/16	8

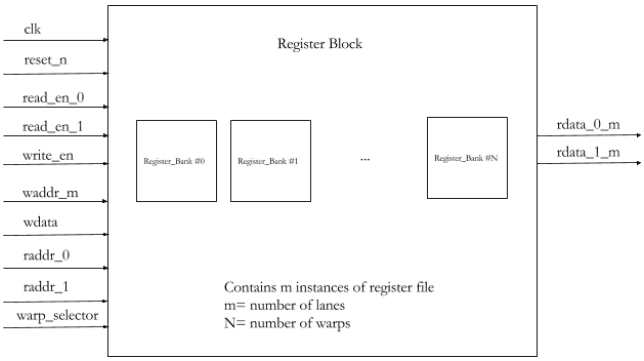

Figure 3: Block Diagram of Register Block

Table 8: I/O Description for Register Bank

Name	Type	Width	Description
clk	input	1	clock source
reset_n	input	1	Active low synchronous reset
read_en_0	input	NUM_LANES	Read from raddr_0
read_en_1	input	NUM_LANES	Read from raddr_1
write_en	input	NUM_LANES	waddr and wdata valid
raddr_0	input	LOG2_NUM_REGS	Read Address 0 for all lanes
raddr_1	input	LOG2_NUM_REGS	Read Address 1 for all lanes
waddr	input	LOG2_NUM_REGS	Write Address for all lanes
wdata_m	input	DATA_WIDTH	Write Data for lane 'm'
rdata_0_m	output	DATA_WIDTH	Read Data for raddr_0
rdata_1_m	output	DATA_WIDTH	Read Data for raddr_1

m = 0, 1, 2 ... (NUM_LANES-1)

Table 9: Register Bank Test Cases and Description

No.	Test Name	Description
1	write_read_all_lanes_i_j_k i= 8, 16 j= 16, 32, 64 k= 32, 64	Writes random data to same register on each lane Read back through different read ports

i is parameter for NUM_LANES
j is parameter for NUM_REGS
k is parameter for DATA_WIDTH

2.2 Predicate Register Block

The structure of this module is similar to that of the general purpose register block, except that the DATA_WIDTH is 1. The hardware implementation involves instantiating a general purpose register block and overriding the DATA_WIDTH parameters for the module and its sub modules. The test cases used to verify the system are enlisted in Table 12.

Table 10: I/O Description for Register Block

Name	Type	Width	Description
clk	input	1	clock source
reset_n	input	1	Active low synchronous reset
read_en_0	input	NUM_LANES	Read from raddr_0
read_en_1	input	NUM_LANES	Read from raddr_1
write_en	input	NUM_LANES	waddr and wdata valid
raddr_0	input	LOG2_NUM_REGS	Read address 0 for all lanes
raddr_1	input	LOG2_NUM_REGS	Read address 1 for all lanes
waddr	input	LOG2_NUM_REGS	Write address for all lanes
wdata_m	input	DATA_WIDTH	Write data for lane 'm'
warp_selector	input	3: For 8 warps 4: For 16 warps	Register bank selector
rdata_0_m	output	DATA_WIDTH	Lane m read data from raddr_0
rdata_1_m	output	DATA_WIDTH	Lane m read data from raddr_1

m = 0, 1, 2 ... (NUM_LANES-1)

Table 11: Register Block Test Cases and Description

No.	Test Name	Description
1	warp_m_write_read_all_lanes_i_j_k m= 8, 16 i= 8, 16 j= 16, 32, 64 k= 32, 64	Writes random data to same register on each lane of a register bank Read back through different read ports

m is 8 or 16 (number of warps)
i is parameter for NUM_LANES
j is parameter for NUM_REGS
k is parameter for DATA_WIDTH

Table 12: Predicate Register Block Tests and Description

No.	Test Name	Description
1	warp_m_write_read_all_lanes_i_j m= 8, 16 i= 8, 16 j= 16, 32, 64	Writes random data to same register on each lane of a register bank Read back through different read ports

m is 8 or 16 (number of warps)
i is parameter for NUM_LANES
j is parameter for NUM_REGS
k is parameter for DATA_WIDTH