core_interface

Interface			
interface_name	port_direction	port_type	port_name
ST_BLK			
	output	logic	s_input
	output	logic	s_output
-end-			
BC_COMMIT			
	input	logic	s_enable
	input	logic	s_cur
	input	logic	s_cur_fcode
	input	logic	s_spec
	inv	TF_PC	RECOV
	inv	TF_PC	CUR
-end-			
CMD_BLK			
	input	logic	c_pause
	input	logic	c_reset
	input	logic	c_clock
	input	logic	c_enable
-end-			
STRUCTURE_QUEUE			
	input	logic	c_pop

	input	logic	d_push_data
	output	logic	s_full
		logic	s_empty
		logic	s_bub
		logic	d_head
		logic	c_push
-end-			
STRUCTURE_DYNQUEUE			
	input	logic	c_pop_index
	output	logic	d_all_data
		Structure_Queue	
-end-			
STRUCTURE_ARR			
	input	logic	c_insert
	input	logic	i_index_input
	input	logic	i_index_output
	input	logic	d_ins_data
	output	logic	d_rtr_data
	output	logic	d_all_data
-end-			
TF_PC			
	input	logic	i_pc
	input	logic	i_pvl
-end-			
TF_ARC_INSTR			
	output	logic	d_instr
-end-			
TF_MARC_INSTR			
	output	logic	i_preg_rd
	output	logic	i_preg_r1
	output	logic	i_preg_r2
	output	logic	s_preg_r1
	output	logic	s_preg_r2
	output	logic	d_preg_r1
	output	logic	d_preg_r2
	output	logic	d_imm
	output	logic	i_creg_r1
	output	logic	i_pip
	output	logic	c_op
		TF_PC	PC
-end-			

BC_PREG			
	input	logic	i_preg_rb1
	input	logic	i_areg_rb1
	input	logic	d_preg_rb1
-end-			
BC_CREG			
	input	logic	i_preg_rb1
	input	logic	d_preg_rb1
-end-			
COM_REGMNG_CALL	[implicit rule]owner is register manager		
	input	logic	i_areg_rd
	input	logic	i_areg_r1
	input	logic	i_areg_r2
	input	logic	c_req
	input	logic	i_spec_pc
	input	logic	i_spec_pvl
	input	logic	i_pip
	input	logic	c_op
	output	logic	i_preg_rd
	output	logic	i_preg_r1
	output	logic	i_preg_r2
	output	logic	s_preg_r1
	output	logic	s_preg_r2
	output	logic	s_req_status
	output	logic	d_preg_r1
	output	logic	d_preg_r2
-end-			

COM_ROB_INIT	[implicit rule]owner is register manager		
	output	logic	i_spec_pc
	output	logic	i_spec_pvl
	output	logic	i_areg_rd
	output	logic	i_pip
	output	logic	c_op
	output	logic	c_req
	input	logic	s_book
	input	logic	i_preg_rd
-end-			
CMD_COMMIT_DIRECTOR			
	output	logic	c_store
	output	logic	c_load
	output	logic	c_ptoa_reg
	output	logic	c_ctoa_reg
	output	logic	c_atoc_reg
-end-			
CMD_ROB_FILL			
	output	logic	i_preg_rd
	output	logic	i_epip
-end-			
FILL_ROB_GLOB			
	output	logic	d_preg_rd
	output	logic	d_creg_r1
	output	logic	i_creg_r1
	output	logic	i_trapc
	inv	TF_PC	SPEC_RES
-end-			
TF_ROB_FILL			
	output	logic	d_preg_rd
	output	logic	i_creg_rc
	output	logic	i_arch_nextPc

core_interface

	output	logic	i_mem_addr
	output	logic	s_trapC
-end-			
-eof-			

description	interface_spec_size	
	G	
ready to accept(1) Refuse(0)	1	
output is occur(1) output is in-compute or dropped(0).	1	
	G	
status that rob is committing the instruction(1) Otherwise (0)	1	
status that current committing instruction is no fault(1), Otherwise (0)	1	
fault code tell cause current instruction can't commit but if enable is not set or current committing instruction isn't fault this value can be anything.	I_BL_TRAPC	
status that tell whether next instruction is speculated for properly address(1). Otherwise is(0). Please not that if current instruction isn't in committing or got a fault declaration this value can be anything	1	
used for transfer correct program counter when s_spec is mispredict.	TP1	
used for transfer current address	TP1	
	G	
pause(stall) all state in the queue and hold the outcome for next cycle	1	
clear all state and register to initial value at next positive edge.	1	
clock signal	1	
order the block to accept the input; 1 is enable, otherwise is 0;	1	
	UNUSED	
pop command data at queue's head.	1	

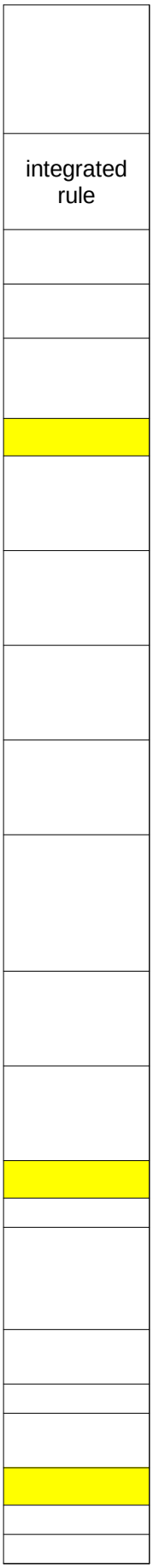
data that intend to push to the queue.	1	
The queue is full.	1	
The queue is empty.	1	
There are at least one bubble at the head of the queue.	1	
data at the queue's head	1	
insert command data at queue's tail.	1	
	UNUSED	
address that user require to bring it out and shift all later queue entry closer by 1 compared to head of queue		
array of data in the queue.		
	UNUSED	
command that intend to insert to array Insert (1) do not insert anythings(0)	1	
G		
index that want to retrieve the data.		
data that we want to insert to array.(position corresponds to i_index_input)		
retrieved data that correspond to i_index_output		
array of all data that laid in the structure		
	TP1	
program counter	I_BL_ARC_PC	
i_pc's privilege machine Mode(0)Hypervision (1)User mode (2)	I_BL_ARC_PVL	
	TRI1	
architecture instruction	D_BL_ARC_INSTR	
	TMI1	
Destination physical register address	I_BL_MARC_REG	
r1 physical register address	I_BL_MARC_REG	
r2 physical register address	I_BL_MARC_REG	
valid index that data for r1 id valid right now.	1	
valid index that data for r2 id valid right now.	1	
data of physical reg r1	D_BL_MARC_REG	
data of physical reg r2	D_BL_MARC_REG	
immediate value that was specified from instruction.	D_BL_MARC_IMM	
control status register address	I_BL_MARC_CREG	
index of reservation station	I_BL_MARC_PIP	
Micro-opcode	D_BL_MARC_OP	
for transfer program counter and privilege.	TP1	

	G	
address/index of physical register that wish to broadcast. [address 0 is implicit rule when did not want to broadcast data]	I_BL_MARC_REG	
address/index of architecture register that preg refer to.	I_BL_ARC_REG	
data of physical register that wish to broadcast.	D_BL_MARC_REG	
	G	
address/index of control status register that wish to broadcast. [address 0 is implicit rule when did not want to broadcast data]	I_BL_MARC_CREG	
data of control status register that wish to broadcast.	D_BL_MARC_CREG	
	RDR1	
destination architecture register[0 is implicit address when don't want this register]	I_BL_ARC_REG	
first architecture register address [0 is implicit address when don't want this register]	I_BL_ARC_REG	
second architecture register address [0 is implicit address when don't want this register]	I_BL_ARC_REG	
1 request reorder buffer and physical register	1	
address of later consecutive instruction that has been speculate.	I_BL_ARC_PC	
previlege of later consecutive instruction that has been speculate.	I_BL_ARC_PVL	
index of reservation station to tell rob	I_BL_MARC_PIP	
Micro-opcode to store to rob	D_BL_MARC_OP	
destination architecture register[0 is implicit address when don't want this register]	I_BL_MARC_REG	
first architecture register address [0 is implicit address when don't want this register]	I_BL_MARC_REG	
second architecture register address [0 is implicit address when don't want this register]	I_BL_MARC_REG	
specify that preg r1 data that give to decoder is valid; 1 is correct	1	
specify that preg r2 data that give to decoder is valid; 1 is correct. If the requestor did no intend to use r2 so the status use be set to 1	1	
status that book for rob is complete and assume that physical address can deliver in one cycle if not register manager should not book for rob(assume that every instruction need to book rob)	1	
first architecture register address [0 is implicit address when don't want this register]	D_BL_MARC_REG	
second architecture register address [0 is implicit address when don't want this register]	D_BL_MARC_REG	

	RRR1	
address of later consecutive instruction that has been speculate.	I_BL_ARC_PC	
previlege of later consecutive instruction that has been speculate.	I_BL_ARC_PVL	
sent to rob to tell what architecture that they should update when this called instruction has been committed.	I_BL_ARC_REG	
index of reservation station to tell rob	I_BL_MARC_PIP	
Micro-opcode to store to rob	D_BL_MARC_OP	
output wire to request area from rob which i_pip is used to specify the rob lane.	1	
book status from rob book complete(1) otherwise(0)	1	
address/index of reorder buffer that correspond with booking if no booking or booking incomplete, the value can be anything.	I_BL_MARC_REG	
	G	
Guild committer to store data in physical register to memory	1	
Guild committer to load data from memory to architectural register	2 bit	
guild committer to store data from physical register to architectural register	3 bit	
guild committer to load data from control status register to architectural register	4 bit	
guild committer to load data from architectural register to control status register	5 bit	
	CMF	
address/index of physical register that wish to store to reorder buffer. [implicit rule address 0 for don't want to store to reorder]	I_BL_MARC_REG	
index of execution pipe in connected rsv which want to store data to[NO IMPLICCIT RULE FOR DUMMY PIPE HERE] rob	I_BL_EX_PIP	
	UNUSED	
data that execution unit which to fill to rob	D_BL_MARC_REG	
data which want to store to control status register	D_BL_MARC_CREG	
address that want to store or req to rob	D_BL_MARC_CREG	
trap cause that these instruction generated.	I_BL_TRAPC	
for describe next pc and privilege that next instruction should be [spec result]	TP1	
	ERF1	
hold data which represent the register value	D_BL_MARC_REG	
address of control status register	I_BL_MARC_CREG	
hold correct next program counter	I_BL_ARC_PC	

how memory address which will be sent to load store unit	I_BL_MEM_ADDR	
hold trap cause for current instruction	I_BL_TRAPC	

			variable convention			
			c_<verb>_<des>	i_<name>	d_<name>	s_<name>


integrated
rule