

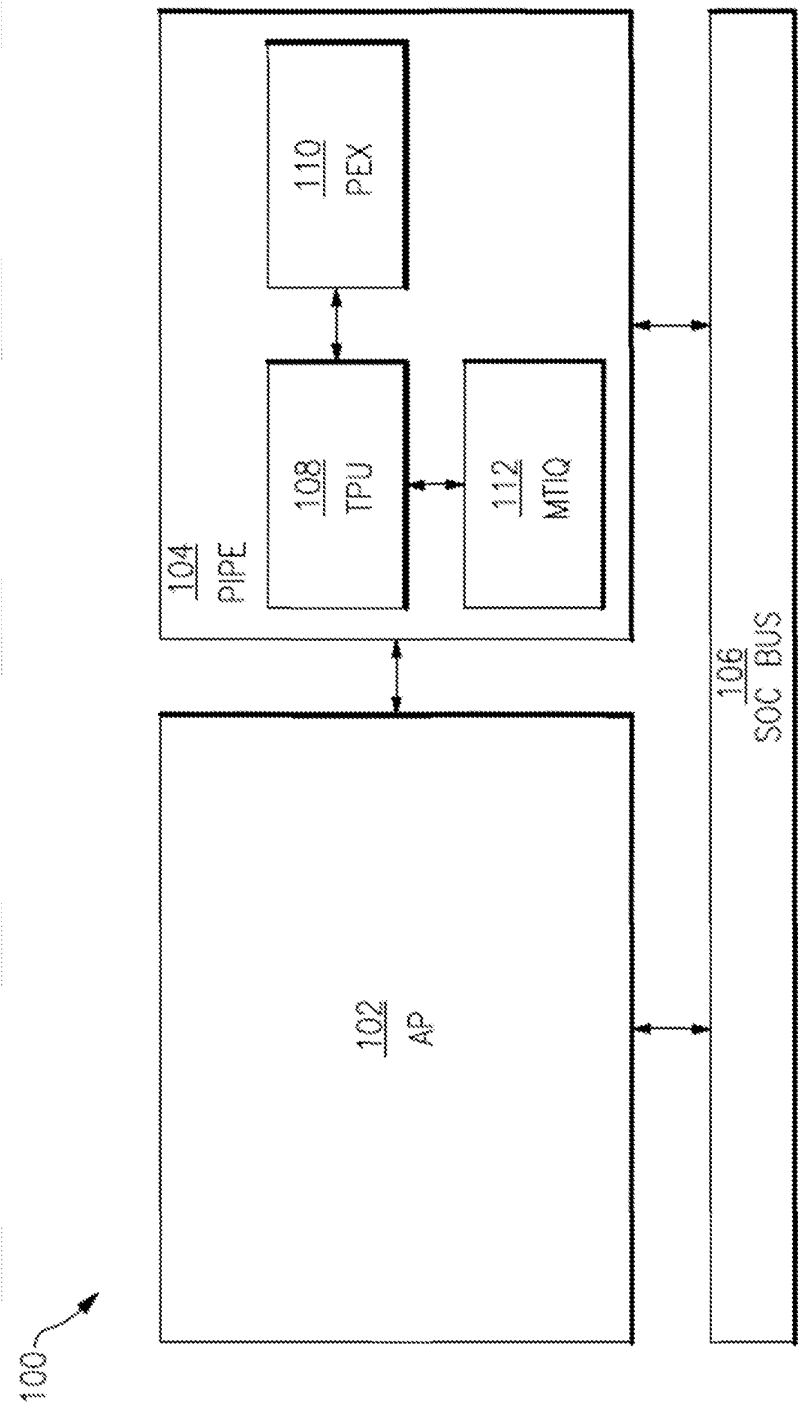


FIG. 1

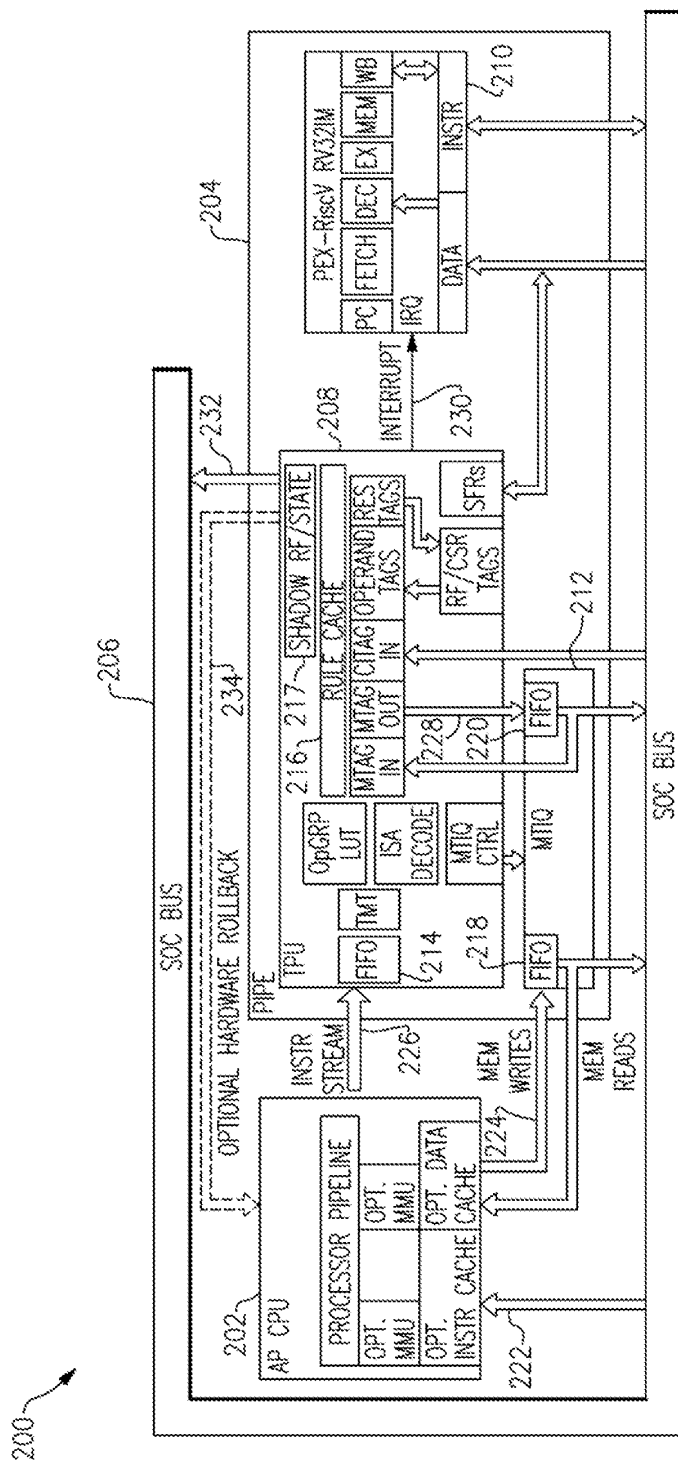


FIG. 2

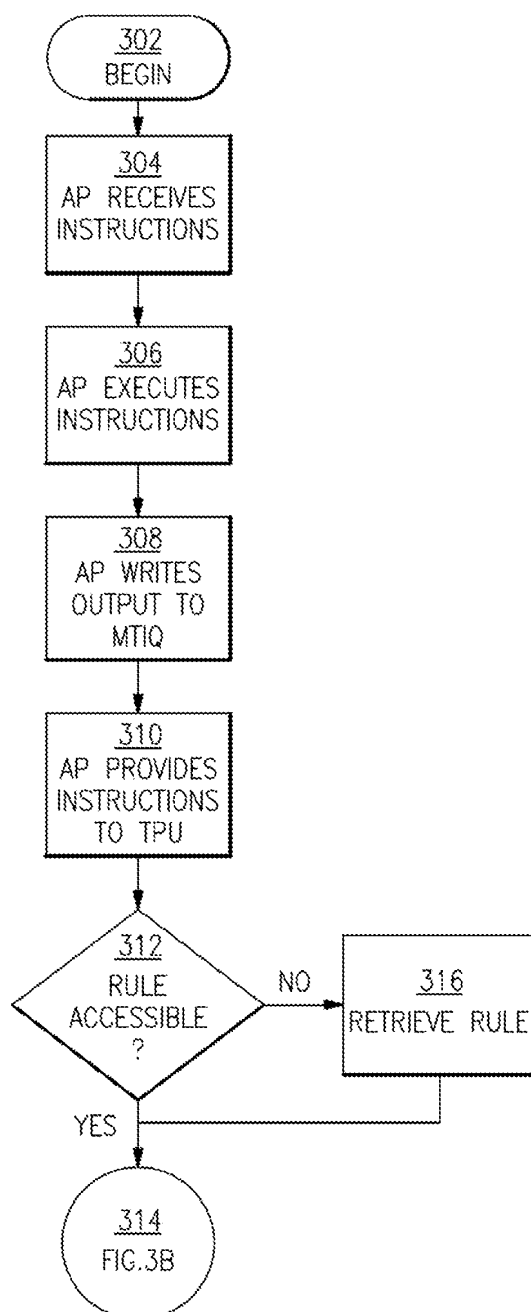


FIG. 3A

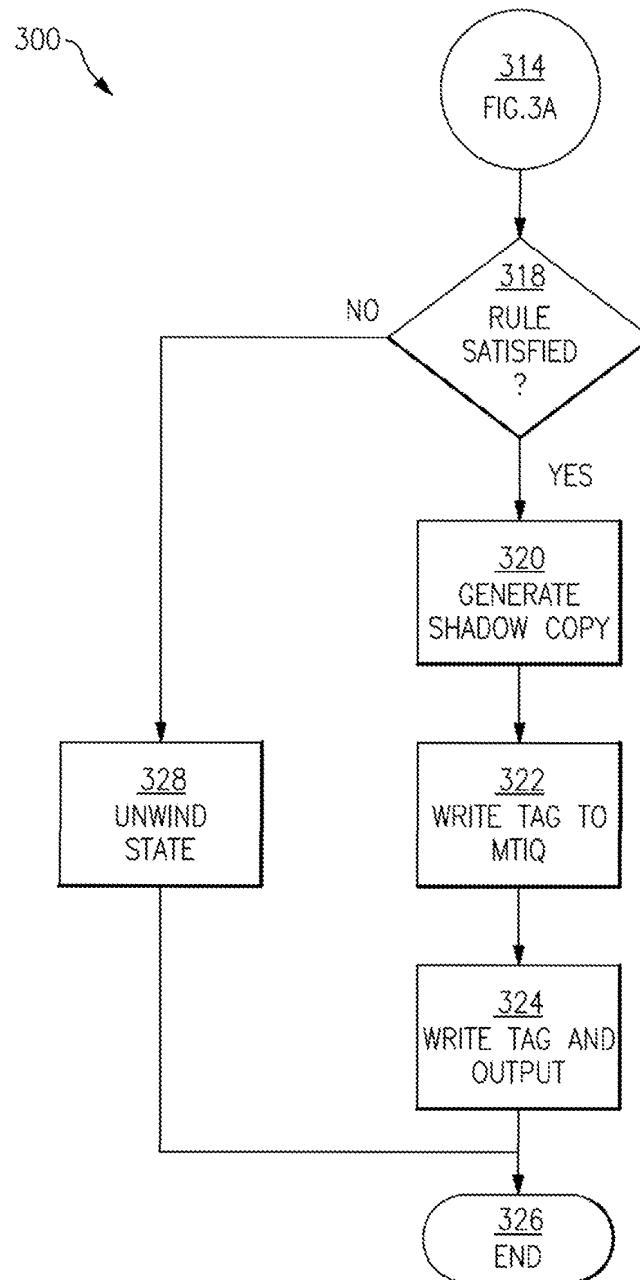


FIG. 3B

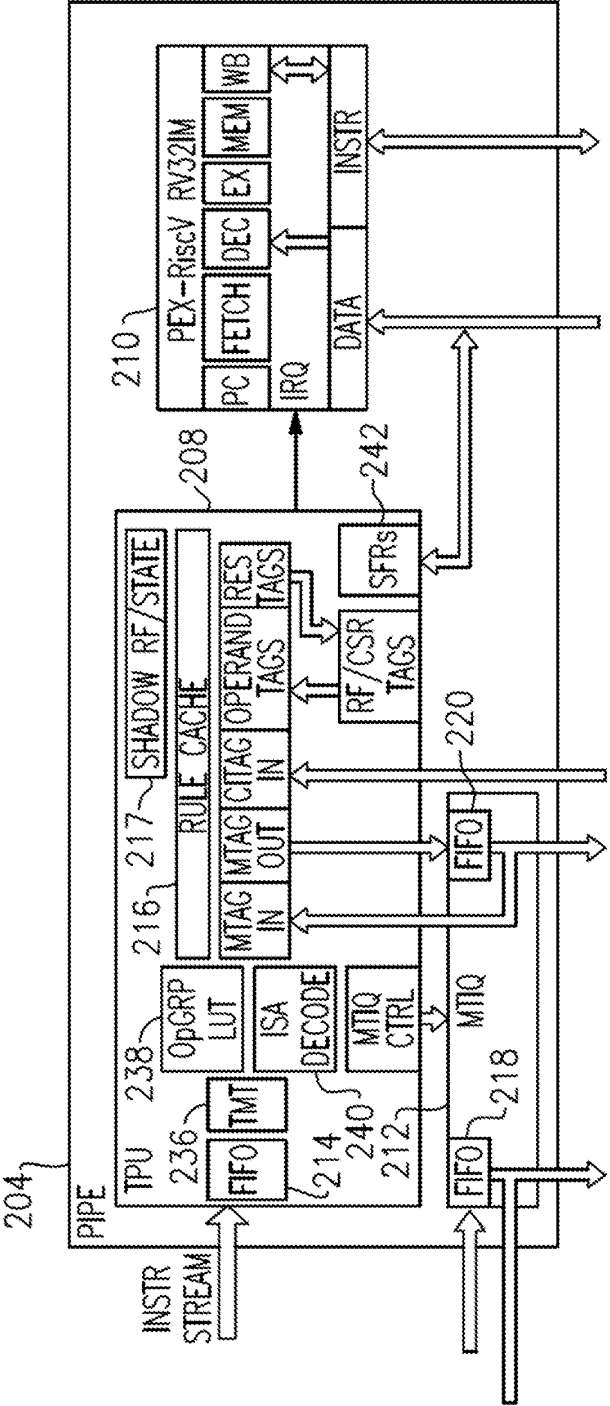


FIG. 4

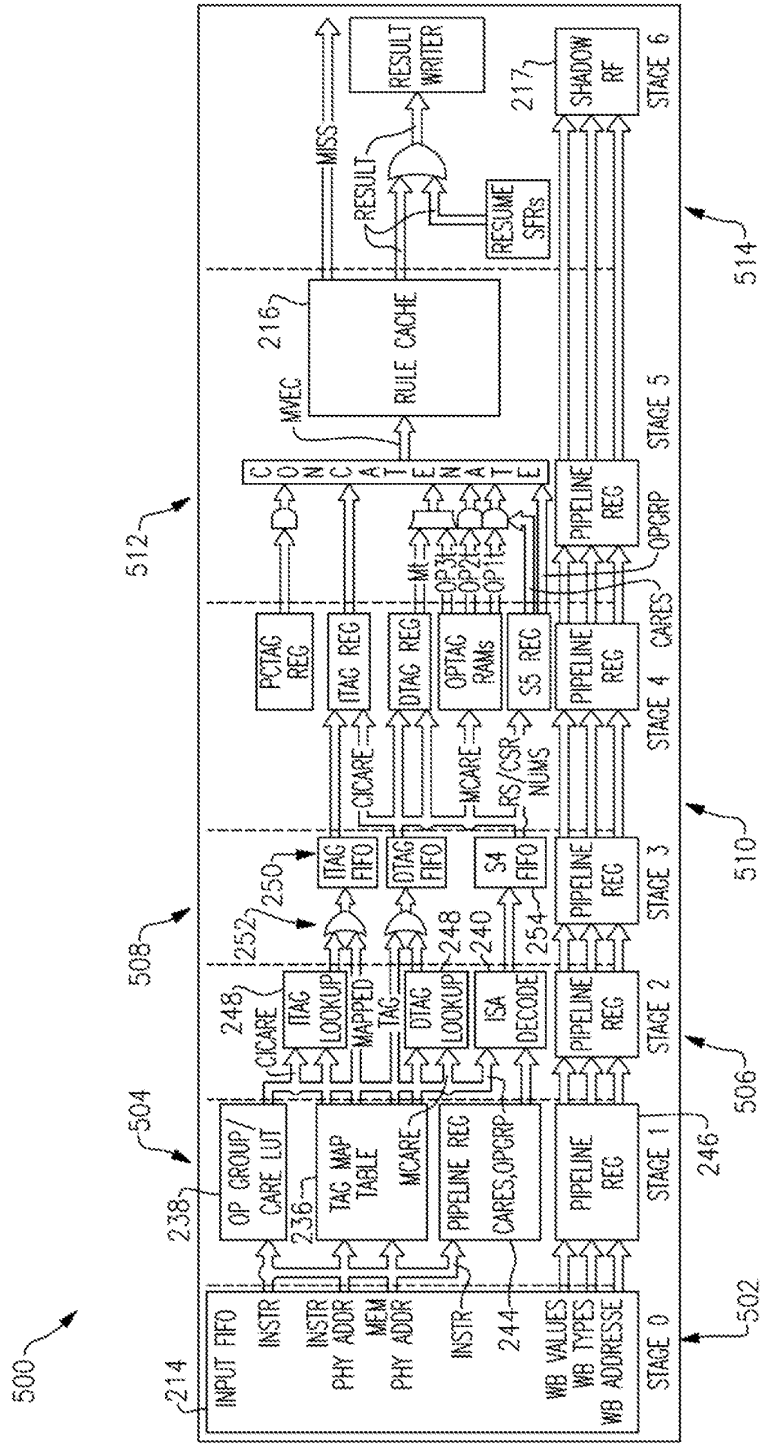


FIG. 5

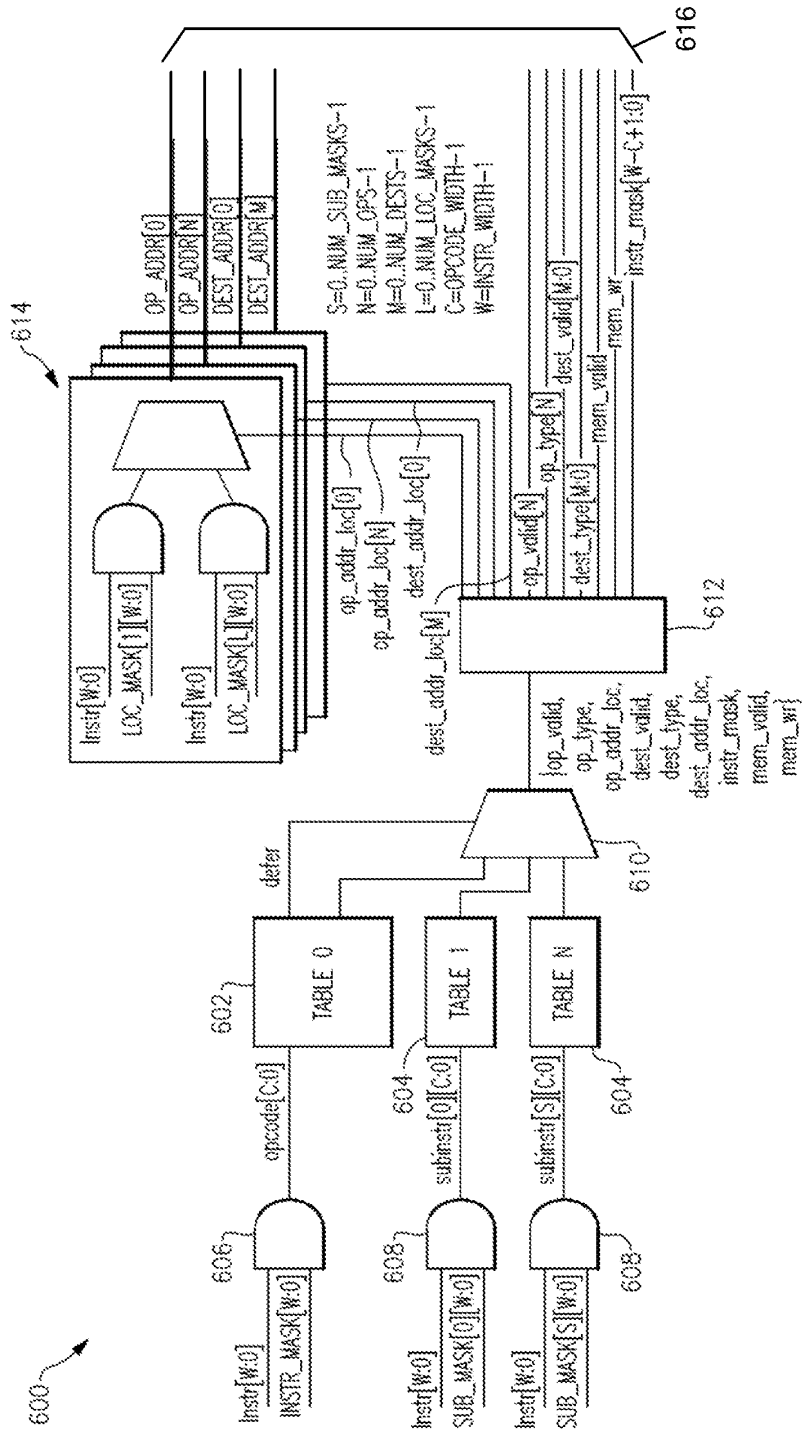


FIG. 6

PIPE Registers

Address	Bit(s)	Name	Access	Description
0x000		PipeState		State of PIPE
	0	RuleMiss	R	A Rule Miss has occurred
	2	TMT_IMiss	R	Tag Map Table had no entry for Instruction Address
	3	TMT_DMiss	R	Tag Map Table had no entry for Data Address
	4	InputFull	R	Input FIFO is full
	5	InputEmpty	R	Input FIFO is empty
0x008	9:0	InputFifoCnt	R	Count of PIPE input FIFO
0x028	31:0	RCMissCnt	R/W	Counter of PIPE Rule Cache Misses
0x030	31:0	RCHitCnt	R/W	Counter of PIPE Rule Cache Hits
0x080	3:0	APIRQ	W/HC	Generates interrupt pulses to the AP (through a Platform Interrupt Controller). Up to 4 interrupts may be generated depending on the bits set in the word written.
0x088		APCtrl	R/W	
	31:2	APMivvec[31:2]	R/W	Sets the reset mvec value to the AP. Bits [1:0] are always zero. Default = 0x100
	0	APResetn	R/W	Put the AP in reset when 0, releases reset when 1. Defaults to 0 to keep AP in reset while PEX boots. Default = 0
0x098	31:0	InputFlush	R/W	Any value written clears PIPE input FIFO and sends interrupt to AP. Automatically clears to zero
0x0A0	31:0	RCFlush	R/W	Write to this address triggers PIPE Rule Cache flush. Automatically clears to zero
0x100		MissInstr		Instruction/Opgroup bits for rule miss
	9:0	MissOpgrp	R	Opgroup bits for rule miss
	31:10	MissInstr	R	Non-opcode instruction bits for rule miss
0x108		MissLSBs	R	LSBs of the instruction and data addresses for a rule miss
	2:0	MissLSBD	R	LSBs of the data address for a rule miss
	5:4	MissLSBI	R	LSBs of the instruction address for a rule miss
0x110	31:0	MissPC	R	PC (Program Counter) tag for rule miss
0x118	31:0	MissCI	R	CI (Current Instruction) tag for rule miss
0x120 0x128 0x130	31:0	MissOP[n]	R	RS[n] (Register Source) tag for rule miss
0x180	31:0	MissDataVA	R	Virtual address of Data Memory accessed for rule miss

FIG. 7A

PIPE Registers (continued)

Address	Bit(s)	Name	Access	Description
0x188	31:0	MissDataPA	R	Physical address of Data Memory accessed for rule miss
0x190	31:0	MissDataTagPA	R	Physical address of Data Memory Tag for rule miss
0x198	31:0	MissInstrVA	R	Virtual Address of current instruction for rule miss
0x1A0	31:0	MissInstrPA	R	Physical Address of current instruction for rule miss
0x1A8	31:0	MissInstrTagPA	R	Physical Address of CI Tag for rule miss
0x200 0x208 0x210 0x218	9:0	MissHash[n]		Address hash of the input key of the miss Value of the first hash of the rule inputs (address into first Rule Cache set), and accounting bits of the entry at that location in the Rule Cache set (valid, aging)
0x240 0x246 0x248 0x256		MissInfo[n]	R	Information about the entry at the address of MissHash[n]
	15:0	MissInstallCnt[n]	R	16-bit installation count from when this rule cache entry was installed. Used by hardware to generate the Least Recently Installed (LRI) candidate for rule installation, which is written into WrWayDest when WrCopy is used.
	23:20	MissScratch[n]	R	Scratch bits for SW
	24	MissLock[n]	R	Lock bit for LRU algorithm
	31	MissValid[n]	R	Flag to indicate the rule cache entry is valid
0x300	31:0	WrInstr	R/W	Instr/OpGroup value to be installed
	9:0	WrOpgrp	R	Opgroup bits value to be installed
	31:10	WrInstr	R	Non-opcode instruction bits value to be installed
0x308		WrLSBs		Address and Data LSBs to be installed
	2:0	WrLSBD	R/W	LSBs of the data address for a rule miss
0x310	31:0	WrPC	R/W	PCtag value to be installed
0x318	31:0	WrCI	R/W	CItag value to be installed
0x320 0x328 0x330	31:0	WrOp[n]	R/W	Op[n] tag value to be installed. Depending on the ISA, may be used for result tags when an instruction has more destinations than WrR#Tags available. Which WrOp[n] entry is used for a particular destination tag depends on the ISA. For RISC-V, for example, CSR instructions use WrOp[3] for the result tag on the CSR.
0x380		WrDest	R/W	

FIG. 7B

PIPE Registers (continued)

Address	Bit(s)	Name	Access	Description
	3:0	WrWayDest	R/W	Indicates which Rule Cache way(s) to install the rule to.
	23:20	WrScratch	R/W	Bits usable by SW for accounting
	24	WrLock	R/W	Lock bit for LRU algorithm
	31	WrInValid	R/W	Marks the entry as invalid when committed. Useful for removing a rule from the cache.
0x388	31:0	WrTransID	R/W	Current transaction id
0x390	31:0	WrCopy	R/W	Triggers a copy of the values in the Miss* registers into the corresponding Wr* registers if the value written here matches the value in WrTransID. The copy happens atomically in a single cycle. During the copy, the HW calculated Least Recently Installed way, or the first invalid way, is written to WrWayDest
0x3A0	31:0	WrCommit	R/W	Triggers write to PIPE Rule Cache if value matches TransID
0x3A8	31:0	WrResume	R/W	Triggers a resume in the PIPE using the results supplied for the missed input values if the value written matches TransID
0x408	31:0	WrRPCTag	R/W	Result tag for the PC tag. Some values are reserved to indicate the result tag is to be copied from an input tag. 0=Writes zeros 1=Copy PC tag 2=Copy CI tag 3=Copy OP1 tag 4=Copy OP2 tag 5=Copy OP3/Memory tag 6+=Writes the value as the tag
0x410	31:0	WrRITag	R/W	Result tag for the first destination of an instruction. Same interpretation as WrRPCTag
0xD00	5:0	TMT_Entry	R/W	Entry Number for write to Tag Map Table plus valid bit and tag bit
0xD08	1:0	TMT_Type	R/W	Specifies Entry type and valid
	0	TMT_IVValid	R/W	Sets the Tag Map Table entry specified in TMT_Entry as valid or invalid for CI Tag address mapping when TMT_Value is written
	1	TMT_DValid	R/W	Sets the Tag Map Table entry specified in TMT_Entry as valid or invalid for Memory Tag address mapping when TMT_Value is written

FIG. 7C

PIPE Registers (continued)

Address	Bit(s)	Name	Access	Description
	1	TMT_DValid	R/W	Sets the Tag Map Table entry specified in TMT_Entry as valid or invalid for Memory Tag address mapping when TMT_Value is written
	2	TMT_Imm	R/W	Indicates that the mapping for this page is always to the tag specified in TMT_Value, not to a memory page to fetch a tag from. This can be used as a performance/power/memory enhancement when an entire page has the same tag, and as a way to supply tags for peripheral devices.
	5:4	TMT_Size	R/W	Indicates the size of the tags to be fetched from memory when this mapping is used. 00 = Byte 01 = Half word (2 bytes) 10 = Word (4 bytes) 11 = Reserved for double word in future
	10:8	TMT_Shift	R/W	Indicates the number of nibbles the input physical address is shifted right after masking before being OR'd with TMT_Val to generate the physical tag address.
0xD10	31:0	TMT_Addr	R/W	Physical Address value being mapped
0xD18	31:0	TMT_Mask	R/W	Mask to apply to physical addresses before compare to TMT_Addr
0xD20		TMT_Val	R/W	Output value for Tag Map Table entry. Writing this register causes a write to the Tag Map Table
	19:0	TMT_Page[19:0]	R/W	When TMT_Imm=0, specified the physical page address of the tag for the page address given in TMT_Entry. When TMT_Imm=1, becomes the lower bits of the tag.
	31:20	TMT_Tag[31:20]	R/W	Upper bits of tag when TMT_Imm=1
0xD28	31:0	TMT_Miss_IPA	R	Address of the instruction physical address that caused the last TMT_IMiss interrupt
0xD30	31:0	TMT_Miss_DPA	R	Address of the memory physical address that caused the last TMT_DMiss interrupt
0xD38	31:0	TMT_Miss_Instr	R	Instruction that generated the last TMT_I_Miss or TMT_D_Miss.
0xE00	0	OpgrpEn	R/W	Enable the use of Op Group/Care table. When disabled, opcode is used directly in the key and all input cares are always set. Default = 0

FIG. 7D

PIPE Registers (continued)

Address	Bit(s)	Name	Access	Description
0xE08	9:0	OpgrpAddr	R/W	Address for write to opgroup/care table.
0xE10		OpGrpValue	R/W	Data for write to opgroup/care table. Causes table write
	9:0	OpGrp	R/W	The group the OpCode+Func3 specified belongs to
	31:10	CareInstr[21:6]	R/W	Instruction care bits for portions of the instruction not used as OpCode
0xE18		OpgrpCares	R/W	Data for write to opgroup/care table
	2:0	CareLSBD	R/W	Care bits for the LSBs of the data address
	4:3	CareLSBI	R/W	Care bits for the LSBs of the instruction address
	5	CarePC	R/W	Care bit for the PC Tag
	6	CareCI	R/W	Care bit for the CI Tag
	7	CareM	R/W	Care bit for the M Tag
	10:8	CareOP[n]	R/W	Care bit for the OP[n] Tag
0x8000 0x8008 ...	31:0	TagMem[n]	R/W	Current tag on AP's Register Files and CSRs. See Table 4 for address mapping of specific ISA.
0x8F00	31:0	PCtag	R/W	Current value of PC Tag
0x8F10	31:0	PCtagSave0	R/W	Value of PC tag at assertion of interrupt event 0
0x8F18	31:0	PCtagSave1	R/W	Value of PC tag at assertion of interrupt event 1
0xC000 0xC008 ...	31:0	ShadowMem[n]	R/W	Shadow value of AP's Register Files and CSRs. Same ISA-specific address mapping as TagMem[n]
0xCF00	31:0	ShadowPC	R	PC of last validated instruction
0xCF10	31:0	ShadowPCSave0	R	PC of instruction aborted in PIPE at assertion of interrupt event 0
0xCF18	31:0	ShadowPCSave1	R	PC of instruction aborted in PIPE at assertion of interrupt event 1

FIG. 7E

LUT Care Bits

Address	Description
9:0	OP Group
31:10	Instruction Care (not including OPCODE bits)
32	<u>PC</u> tag Care
33	<u>CI</u> tag Care
34	<u>M</u> tag Care
36:35	<u>Instr LSB</u> [1:0] (bit 1 is meaningless for a 32-bit system). Enables support of sub-instructions for systems with multiple instructions in a single native word.
39:37	Data <u>LSB</u> [2:0] cares (bit 2 is meaningless in a 32-bit system). Multiple its enable specifying exactly how many <u>LSBs</u> the <u>OPGRP</u> uses.
40	<u>OP1</u> tag Care
41	<u>OP2</u> tag Care
42	<u>OP3</u> tag Care

FIG. 8

TPU Communication Parameters

Parameter Name	Value	Description
<u>RESET_POL</u>	1'b0	Polarity of asserted reset. 1'b0 indicates active-low reset.
<u>CACHE_WAY_DEPTH</u>	1024	Depth of each Rule Cache Way
<u>CACHE_WAYS</u>	4	Number of Rule Cache Ways
<u>PA_WIDTH</u>	32	Width of physical address
<u>DATA_WIDTH</u>	32	Width of native word size
<u>TAG_WIDTH</u>	32	Width of tags
<u>IN_FIFO_DEPTH</u>	128	Depth of input FIFO from CPU
<u>MEM_FIFO_DEPTH</u>	16	Depth of Stage 3 FIFOs
<u>NUM_IRQ</u>	4	Number of <u>IRQs</u> that the <u>PEX</u> can generate via the <u>TPU's SFRs</u>

FIG. 9A

TPU Communication Interfaces

Interfaces	Partner	Description
<u>Bus_cpu_tpu</u>	AP CPU Core	Stream of committed instructions and related data from the CPU core.
<u>Bus_inmem</u>	<u>AXI</u> Fabric	Memory access port for reading CI Tags
<u>Bus_dmem</u>	<u>MTIQ</u>	Memory access port for reading Memory Tags
<u>Bus_sfrs</u>	<u>AXI</u> Fabric	Target <u>AXI</u> interface for <u>TPU SFRs</u> and shadow RAMs
<u>Bus_mtiq</u>	<u>MTIQ</u>	Simple interface to send tag writes to <u>MTIQ</u>

FIG. 9B

TPU Communication Interface Inputs

Inputs	Source	Description
Clock	Cpu-clk	CPU Clock
Reset	Power-on Reset	Reset signal
imem_id	Constant at SOC top	AXI ID for TPU fetches of CI tags
dmem_id	Constant at SOC top	AXI ID for TPU fetches of Data tags
Non_tag_addr	Constant at SOC top	Special address sent to MTIQ when memory write used an immediate tag from TMT

FIG. 9C

TPU Communication Interface Outputs

Outputs	Description
Pipe_pex_irq	Interrupt request to PEX to service misses
Ap_nrst	Active-low reset to AP
Ap_tvec	Reset vector to AP

FIG. 9D

TPU ISA Parameters

Parameter Name	Value	Description
INSTR_WIDTH	32	Width of instruction word
OPCODE_WIDTH	10	Number of Opcode bits used to address operation group/care LUT
OPGRP_WIDTH	10	Width of OPGRP output
OP_ADDR_WIDTH	12 (for CSR)	Max size address of all operand types
NUM_OPS	3	Max number of operands in any one instruction
NUM_OP_TYPES	3 (CSR, FP KE, Integer RF)	Number of independent operand types
NUM_DESTS	3 (CSR & RD on CSR Swapr)	Max number of destinations in an instruction
NUM_OP_TAGS	130	Number of unique operand tags needed for all operand types in the CPU
DED_MR_KEY_SLOT	0	Indicates if the tag read from Memory has a dedicated slot in the rule key, or if it shares a slot with the last operand number (0=shared, 1=dedicated)
INSTR_OPCODE_MASK	32'h0000_707F (funct3+Opcode)	Mask to specify position of Opcode bits in instruction.

FIG. 10

Address Map for TagMem and ShadowMem

PIPE TagMem / ShadowMem Offset	AP Register File/CSR Name	AP RF/CSR Address
0x0 -- 0x7F	Integer Registers 0-31	0-31
0x80	CSR fflags, frm , fcsr	0x001-0x003
0x84	CSR cucle, cucleh, cyclew, cyclehw	0xC00, 0xC80, 0x900, 0x980
0x88	CSR time, timeh, timew, timehw	0xC01, 0xC81, 0x901, 0x981
0x8C	CSR instret, instreth, instretw, instrethw	0xC02, 0xC82, 0x902, 0x982
0x90	CSR sstatus	0x100
0x94	CSR stvec	0x101
0x98	CSR sie	0x104
0x9C	CSR stimecmp	0x121
0xA0	CSR stime, stimeh, stimew, stimehw	0xD01, 0xD81, 0xA01, 0xA81
0xA4	CSR sscratch	0x140
0xA8	CSR sepc	0x141
0xAC	CSR scause	0xD42
0xB0	CSR sbadaddr	0xD43
0xB4	CSR sip	0x144
0xB8	CSR spfbr	0x180
0xBC	CSR sasid	0x181
0xC0	CSR mcuid	0xF00
0xC4	CSR mimpid	0xF01
0xC8	CSR mhartid	0xF10
0xCC	CSR mstatus	0x300
0xD0	CSR mtvec	0x301
0xD4	CSR mideleg	0x302
0xD8	CSR mie	0x304
0xDC	CSR mtimecmp	0x321
0xE0	CSR mtime	0x701
0xE4	CSR mtimeh	0x741
0xE8	CSR mscratch	0x340
0xEC	CSR mepc	0x341
0xF0	CSR mcause	0x342
0xF4	CSR mbadaddr	0x343
0xF8	CSR mip	0x344
0xFC	CSR mbase	0x380
0x100	CSR mbound	0x381
0x104	CSR mibase	0x382
0x108	CSR mibound	0x383
0x10C	CSR mdbase	0x384
0x110	CSR mdbound	0x385

FIG. 11

MTIQ Interface Parameters

Parameter Name	Value	Description
<u>PA_WIDTH</u>	32	Width of physical address
<u>BUS_WIDTH</u>	32	Width of system's <u>AXI</u> bus
<u>TAG_WIDTH</u>	32	Width of tags
<u>FIFO_DEPTH</u>	128	Depth of input FIFO from CPU

FIG. 12A

MTIQ Interfaces

Interfaces	Partner	Description
<u>Bus_cpu_pipe</u>	AP CPU Core	Stream of committed instructions and related data from the CPU core
<u>Bus_cpu_mtiq</u>	CPU Data <u>AXI</u> Interface	CPU's data read and write <u>AXI</u> port
<u>Bus_pipe_mtiq</u>	PIPE Data <u>AXI</u> Interface	<u>PIPE</u> 's data tag read and write <u>AXI</u> port
<u>Bus_cpu_dmem</u>	<u>AXI</u> Fabric	<u>AXI</u> port for CPU reads, and <u>MTIQ</u> data writes on behalf of CPU
<u>Bus_pipe_dmem</u>	<u>AXI</u> Fabric	<u>AXI</u> port for PIPE reads and <u>MTIQ</u> data writes on behalf of PIPE

FIG. 12B

MTIQ Interface Inputs

Inputs	Source	Description
Clock	Cpu_clk	CPU clock
Reset	Power-on Reset	Power on reset
Invalid_tag_value	PIPE SFRs	The special tag value to be written in the first stage of an MTIQ write
Non_tag_addr	PIPE SFRs	The special address that indicates a write is allowed without a corresponding tag write

FIG. 12C