Model's description

The following list describe the basic behavior for each of the models defined in the library. The model's name is included, and also the kind of operation and input/output ports used.

ADDER

It adds two operands. Changes the Carry bit according with the result.

Name	In	Out	Description
OPA0-OPA31	X		Op A
OPB0-OPB31	X		Op B
RES0-RES31		X	Result
C		X	Carry

Res = OpA + OpB

Stored as: ATOMIC\ADDER.

ALIGNL

It is used to align the data read in a Load operation.

Name	In	Out	Description
OP0-OP31	X		Op
SIZE0-SIZE1	X		
A0-A1	X		Address lower bits
SIGN	X		
RES0-RES31		X	Result

Size coding: 00: Word, 01: Byte, 10: Half Word

If Size = Word, Res = Op
If Size = Half Word, Res = Op >> (16 * (1 - A_1)) AND 0xFFFF
    If Sign = 1, the bit No. 15 of Res is copied into bits 16-31
If Size = Byte, Res = Op >> (8 * (3 - A_1A_0)) AND 0x00FF
    If Sign = 1, the bit No. 7 of Res is copied into bits 8-31

Stored as: ATOMIC\ALIGNL.

ALIGNS

It is used to align the data to be recorded in a Store operation.

Name	In	Out	Description
OP0-OP31	X		Op
SIZE0-SIZE1	X		
A0-A1	X		Lower bits of address
RES0-RES31		X	Result

Size coding: 00: Word, 01: Byte, 10: Half Word

If Size = Word, Res = Op
If Size = Half Word, Res = Op << (16 * (1 - A_1))
If Size = Byte, Res = Op << (8 * (3 - A_1A_0))

Stored as: ATOMIC\ALIGNL.

ALU

Aritmetic-Logic Unit.

Name	In	Out	Description
OPA0-OPA31	X		Op A
OPB0-OPB31	X		Op B
CIN	X		Carry in
FCOD0-FCOD3	X		Function Code
RES0-RES31		X	Result
C		X	Carry
N		X	Negative
Z		X	Zero
V		X	Overflow

FCod		Res
0000	ADD	OpA + OpB
0001	AND	OpA $\wedge$ OpB
0010	OR	OpA $\vee$ OpB
0011	XOR	OpA XOR OpB
0100	SUB	OpA – OpB
0101	ANDN	~(OpA $\wedge$ OpB)
0110	ORN	~(OpA $\vee$ OpB)
0111	XNOR	~(OpA XOR OpB)
1000	ADDX	OpA + OpB + Cin
1100	SUBX	OpA – OpB – Cin

Boolean operations return 0 in the bits C and O.

Stored as: ATOMIC\ALU.

AND

Boolean gate representing a boolean AND

Stored as: ATOMIC\AND.

BUS

Name	In	Out	Description
DATA0-DATA31	X	X	Data
A0-A31	X	X	Address
BSEL0-BSEL3	X	X	Byte Select
CLOCK	X	X	Clock
AS	X	X	Address Strobe
RD/WR	X	X	Read or Write
DTACK	X	X	Data Acknowledge
ERR	X	X	Error
RESET	X	X	Reset
IRQ1-IRQ15	X	X	Interrupt Request
BUSY	X	X	Bus Busy

Every signal is copied from iput to output.

Stored as: ATOMIC\BUS.

CCLOGIC

Used to decide if a jump must be executed in conditional branches.

Name	In	Out	Description
C	X		Carry
N	X		Negative
Z	X		Zero
V	X		Overflow
COND0-COND3	X		Cond
RES		X	Result

Cond		Description	Res
0000	N	Never	0
0001	E	Equal	Z
0010	LE	Less or Equal	Z ∨ (N XOR V)
0011	L	Less	N XOR V
0100	LEU	Less or Equal Unsigned	C ∨ Z
0101	CS	Carry Set	C
0110	NEG	Negative	N
0111	VS	Overflow Set	V
1000	A	Always	1
1001	EN	Not Equal	~Z
1010	G	Greater	~(Z ∨ (N XOR V))
1011	GE	Greater or Equal	~(N XOR V)
1100	GU	Greater Unsigned	~(C ∨ Z)
1101	CC	Carry Clear	~C
1110	POS	Positive	~N
1111	VC	Overflow Clear	~V

Stored as: ATOMIC\CCLOGIC.

CLOCK

Name	In	Out	Description
CLCK		X	Clock

The clock period must be defined by the modeler.

Stored as: ATOMIC\CLOCK.

CMP

It is used in the Address Unit to see that the addresses do not surpass the Limit register.

Name	In	Out	Description
OPA0-OPA31	X		Op A
OPB0-OPB31	X		Op B
EQ		X	Equal
LW/GT		X	lower/not greater

EQ = (Op A = Op B)
LW = (Op A < Op B)

Stored as: ATOMIC\CMP.
Stored as: COUPLED\CMP. This version is built using digital logic concepts. Each atomic model is a boolean gate, and they are coupled in this model.

CS

Chip selector, used to select one register window according to the address.

Name	In	Out	Description
A0-A31	X		Address
AS	X		Address Strobe
CS		X	Chip Select

It can be set to be associated with a range of addresses.
I.e.: Range = [0xFFFF0000..0xFFFFFFFF]

$CS = AS \wedge A \in Range$

Stored as: ATOMIC\CS.
Stored as: COUPLED\CS. This version is built using digital logic concepts. Each atomic model is a boolean gate, and they are coupled in this model.

CWPLOGIC

It decieds if the global registers or the registers window are accessed.

Name	In	Out	Description
CWP0-CWP4	X		Current Window Pointer
SEL0-SEL4	X		Select

GSEL0-GSEL2		X	Global Select
RSEL0-RSEL8		X	Register Select
R/G		X	Register/Global

If Sel < 8, GSel = Sel and R = 0
else, RSel = (CWP * 16 + Sel - 8) % 512 and R = 1

Stored as: ATOMIC\CPWLOGIC.

INC4

Used to update the nPC register.

Name	In	Out	Description
OP0-OP31	X		Op
RES0-RES31		X	Result = Op + 4

Stored as: ATOMIC\INC4.

INC/DEC

Used to increment/decrement the CWP register.

Name	In	Out	Description
OP0-OP4	X		Op
FCOD	X		1 = INC ; 0 = DEC
RES0-RES4		X	Result

Stored as: ATOMIC\NOT.

IRQLOGIC

It is used to recognize the arrival of an interrupt request.

Name	In	Out	Description	Prioridad	Trap Type
IRQ1-IRQ15	X		Interrupt request	32-IRq	0x11-0x1F
PIL0-PIL3	X		Processor interrupt level		
TF		X	Trap found		
TT0-TT7		X	Trap type		

If any IRq > PIL is on, the one with the higher priority is chosen, and TF is turned on.

Stored as: ATOMIC\IRQLOGIC.

LATCH

Name	In	Out	Description
IN0-IN31	X		
EIN	X		Enable Input
CLEAR	X		

OUT0-OUT31		X	

Intermediate register. If we call L to the value that it contains, then,
If EIn = 1, L = In
If Clear = 1, L = 0
Out = L

Stored as: ATOMIC\LATCH.

MEM

It is used as main storage (memory).

Name	In	Out	Description
DATA0-DATA31	X	X	Data
A2-A31	X		Address
BSEL0-BSEL3	X		Byte Select
AS	X		Address Strobe
RD/WR	X		Read or Write
DTACK		X	Data Acknowledge
ERR		X	Error
RESET	X		Reset

The size can be chosen, and its initial image can be loaded from a file.

Fi AS=1, then read A
If Rd=1
    BSel(M[A]) = BSel(Data)
else
    Data = M[A]
DtAck = 1 when the operation is finished
Err = 1 if the address does not exist
BSel is used to define which bytes in the word are updated (i.e., if BSel = 0011 only the 16 least significative bits are updated).
Reset restores the initial memory image.

Stored as: ATOMIC\MEM.

MUL/DIV

MULtiplier/DIVider.

Name	In	Out	Description
OPA0-OPA31	X		Op A
OPB0-OPB31	X		Op B
YIN0-YIN31	X		Y In
YOUT0-YOUT31		X	Y Out
FCOD0-FCOD1	X		Function Code
RES0-RES31		X	Result

N		X	Negative
Z		X	Zero
DIV_ZERO		X	Division by zero

FCod

00	UMUL	YOut * 2^{32} + Res = OpA × OpB (Unsigned)
01	SMUL	YOut * 2^{32} + Res = OpA × OpB (Signed)
10	UDIV	Res = YIn * 2^{32} + OpA / OpB (Unsigned)
11	SDIV	Res = YIn * 2^{32} + OpA / OpB (Signed)

Stored as: ATOMIC\MULDIV.

MUX

Two input multiplexor.

Name	In	Out	Description
A0-A31	X		Op A
B0-B31	X		Op B
SELA/SELB	X		Select A/Select B
OUT0-OUT31		X	Out

If SelA=1, Out = OpA
else, Out = OpB

Stored as: ATOMIC\MUX.

MUXn

N-input multiplexor. The number of inputs is specified using the parameter nports. The number of bits in each input is specified using the parameter nbits.

Name	In	Out	Description
A0-A*nbits*	X		Op A
B0-B*nbits*	X		Op B
C0-C*nbits*	X		Op C
…	X		…
x0-x*nbits*	X		Op x
SELA	X		Select A
SELB	X		Select B
SELC	X		Select C
…	X		…
SELx	X		Select x
OUT0-OUT*nbits*		X	Out

If SelA=1, Out = OpA
If SelB=1, Out = OpB
If SelC=1, Out = OpC
If SelD=1, Out = OpD

Stored as: ATOMIC\MUXN.

NOT

Boolean gate representing a boolean NOT

Stored as: ATOMIC\NOT.

OR

Boolean gate representing a boolean OR

Stored as: ATOMIC\OR.

REGBLOCK

Register window.

Name	In	Out	Description
ASEL0-ASEL8	X		Select A
BSEL0-BSEL8	X		Select B
CSEL0-CSEL8	X		Select C
CEN	X		Enable C
RESET	X		Reset
AOUT0-AOUT31		X	Output A
BOUT0-BOUT31		X	Output B
CIN0-CIN31	X		Input C

It is built as a block of 512 registers (R)
AOut = R[ASel]
BOut = R[BSel]
If CEn = 1, R[CSel] = Cin
Reset erases the contents of all the registers.

Stored as: ATOMIC\REGBLOCK.

REGGLOB

Global registers.

Name	In	Out	Description
ASEL0-ASEL3	X		Select A
BSEL0-BSEL3	X		Select B
CSEL0-CSEL3	X		Select C
CEN	X		Enable C
RESET	X		Reset
AOUT0-AOUT31		X	Output A
BOUT0-BOUT31		X	Output B
CIN0-CIN31	X		Input C

8-register block. If we call it G, then:
AOut = G[ASel]
BOut = G[BSel]

If CEn = 1 and CSel > 0, G[CSel] = Cin
Reset clears all the registers.

Note: G[0] is always 0.

Stored as: ATOMIC\REGGLOB.

SHIFTER

Shifts the input register to the left or to the right.

Name	In	Out	Description
OPA0-OPA31	X		Op A
OPB0-OPB4	X		Op B
FCOD0-FCOD1	X		Function Code
RES0-RES31		X	Result

Fcod		Res
01	SLL	OpA << OpB
10	SRL	OpA >> OpB (Unsigned)
11	SRA	OpA >> OpB (Signed)

Stored as: ATOMIC\SHIFTER.

SIGNEXTn

It is used to extend the sign of an operand of *nbits* to 32 bits.

Name	In	Out	Description
OP0-OP*nbits*	X		Op
RES0-RES31		X	Result = signextend(Op)

Stored as: ATOMIC\SIGNEXTN.

TRAPLOGIC

It is in charge of managing the traps.

Name	In	Out	Description	Prioridad	Trap Type
INST_ACC_EXCEP	X		Instruction access exception	5	0x01
ILLEG_INST	X		Illegal instruction	7	0x02
PRIV_INST	X		Privileged instruction	6	0x03
WIN_OVER	X		Window overflow	9	0x05
WIN_UNDER	X		Window underflow	9	0x06
ADDR_NOT_ALIGN	X		Address not aligned	10	0x07
DATA_ACC_EXCEP	X		Data access exception	13	0x09
INST_ACC_ERR	X		Instruction access error	3	0x21
DATA_ACC_ERR	X		Data access error	12	0x29
DIV_ZERO	X		Division by zero	15	0x2A

DATA_ST_ERR	X		Data store error	2	0x2B
TRAP_INST	X		Trap instruction	16	0x80+TN
TN0-TN6	X		Trap number		
TF		X	Trap found		
TT0-TT7		X	Trap type		

If any of the inputs (excepting TN) is turned on, the input with the highest priority is chosen, and TF is turned on.

Stored as: ATOMIC\TRAPLOGIC.

WIMCHECK

It is used to see if there is a window overflow/underflow in a save/restore operation.

Name	In	Out	Description
CWP0-CWP4	X		Current Window Pointer
WIM0-WIM31	X		Window Invalid Mask
RES		X	Res = WIM_{CWP}

Stored as: ATOMIC\WIMCHECK.

OR

Boolean gate representing a boolean OR

Stored as: ATOMIC\OR.

LIST OF PARTICIPATING STUDENTS

Main participants

. Alejandro Tróccoli
. Sergio Daicz
. Sergio Zlotnik

Other participating students
. Carlos Diuk
. Barrionuevo
. Calvo
. Corvetto
. Grinberg, Deborah
. Núñez Cortes, Marcos
. Petronio, Federico
. Bender, Ariel
. Brignardello, Darío
. Barletta, Patricio
. Enrique, Sebastian
. Rubinstein, Diego
. Altman, Daniel
. Glinsky, Ezequiel
. Wassermann, Demian
. Pablo L. Cremona
. Pablo S. Sor