

PU-OR1K WIKI

A Processing Unit (PU) is an electronic system within a computer that carries out instructions of a program by performing the basic arithmetic, logic, controlling, and I/O operations specified by instructions. Instruction-level parallelism is a measure of how many instructions in a computer can be executed simultaneously. The PU is contained on a single Metal Oxide Semiconductor (MOS) Integrated Circuit (IC).

The OpenRISC implementation has a 32/64 bit Microarchitecture, 5 stages data pipeline and an Instruction Set Architecture based on Reduced Instruction Set Computer. Compatible with Wishbone Bus. Only For Researching.

FRONT-END Open Source Tools

Verilator

Hardware Description Language SystemVerilog Simulator

```
git clone http://git.veripool.org/git/verilator
```

```
cd verilator
autoconf
./configure
make
sudo make install

cd sim/verilog/regression/wb/vtor
source SIMULATE-IT

cd sim/verilog/regression/ahb3/vtor
source SIMULATE-IT
```

Icarus Verilog

Hardware Description Language Verilog Simulator

```
git clone https://github.com/steveicarus/iverilog
```

```
cd iverilog
./configure
make
sh autoconf.sh
sudo make install

cd sim/verilog/regression/wb/iverilog
source SIMULATE-IT
```

```
cd sim/verilog/regression/ahb3/iverilog
source SIMULATE-IT
```

GHDL

Hardware Description Language GHDL Simulator

```
git clone https://github.com/ghdl/ghdl
```

```
cd ghdl
./configure --prefix=/usr/local
make
sudo make install

cd sim/vhdl/regression/wb/ghdl
source SIMULATE-IT

cd sim/vhdl/regression/ahb3/ghdl
source SIMULATE-IT
```

Yosys-ABC

Hardware Description Language Verilog Synthesizer

```
git clone https://github.com/YosysHQ/yosys
```

```
cd yosys
make
sudo make install

cd synthesis/yosys
source SYNTHESIZE-IT
```

BACK-END Open Source Tools

Back-End Workflow

```
mkdir qflow
cd qflow

git clone https://github.com/RTimothyEdwards/magic
git clone https://github.com/rubund/graywolf
git clone https://github.com/The-OpenROAD-Project/OpenSTA
git clone https://github.com/RTimothyEdwards/qrouter
git clone https://github.com/RTimothyEdwards/irsim
git clone https://github.com/RTimothyEdwards/netgen
```

```
git clone https://github.com/RTimothyEdwards/qflow
```

```
cd magic  
./configure  
make  
sudo make install
```

```
cd graywolf  
mkdir build  
cd build  
cmake ..  
make  
sudo make install
```

```
cd OpenSTA  
mkdir build  
cd build  
cmake ..  
make  
sudo make install
```

```
cd qrouter  
./configure  
make  
sudo make install
```

```
cd irsim  
./configure  
make  
sudo make install
```

```
cd netgen  
./configure  
make  
sudo make install
```

```
cd qflow  
./configure  
make  
sudo make install
```

```
cd synthesis/qflow  
source FLOW-IT
```

for WINDOWS users!

open Microsoft Store and install Ubuntu

type:

```
sudo apt update upgrade
```

```
sudo apt-get install bison cmake flex freeglut3-dev libcairo2-dev libgsl-dev \
libncurses-dev libx11-dev m4 python-tk python3-tk swig tcl tcl-dev tk-dev tcsh
```

FRONT-END

type:

```
sudo apt install verilator
```

```
sudo apt install iverilog
```

```
sudo apt install ghdl
```

```
sudo apt install yosys
```

BACK-END

type:

```
mkdir qflow
```

```
cd qflow
```

```
git clone https://github.com/RTimothyEdwards/magic
```

```
git clone https://github.com/rubund/graywolf
```

```
git clone https://github.com/The-OpenROAD-Project/OpenSTA
```

```
git clone https://github.com/RTimothyEdwards/qrouter
```

```
git clone https://github.com/RTimothyEdwards/irsim
```

```
git clone https://github.com/RTimothyEdwards/netgen
```

```
git clone https://github.com/RTimothyEdwards/qflow
```

```
cd magic
```

```
./configure
```

```
make
```

```
sudo make install
```

```
cd graywolf
```

```
mkdir build
```

```
cd build
```

```
cmake ..
```

```
make
```

```
sudo make install
```

```
cd OpenSTA
mkdir build
cd build
cmake ..
make
sudo make install
```

```
cd qrouter
./configure
make
sudo make install
```

```
cd irsim
./configure
make
sudo make install
```

```
cd netgen
./configure
make
sudo make install
```

```
cd qflow
./configure
make
sudo make install
```

Basic parameters

Parameter	Description	Default	Values
OPTION_OPERAND_WIDTH	CPU data and address widths	32	32, 64
OPTION_CPU0	CPU pipeline core	CAPPUCCINO	CAPPUCCINO
OPTION_RESET_PC	Program Counter upon reset	0x100	N

Caching parameters

Parameter	Description	Default	Values
FEATURE_DATACACHE	Enable memory access data caching	NONE	ENABLED
OPTION_DCACHE_BLOCK_WIDTH	Address width of a cache block	5	n
OPTION_DCACHE_SET_WIDTH	Set address width	9	n

Parameter	Description	Default	Values
OPTION_DCACHE_WAYS	Number of blocks per set	2	n
OPTION_DCACHE_LIMIT_WIDTH	Maximum address width	32	n
OPTION_DCACHE_SNOOP	Bus snooping for cache coherency	NONE	ENABLED
FEATURE_INSTRUCTIONCACHE	Memory access instruction caching	NONE	ENABLED
OPTION_ICACHE_BLOCK_WIDTH	Address width of a cache block	5	n
OPTION_ICACHE_SET_WIDTH	Set address width	9	n
OPTION_ICACHE_WAYS	Number of blocks per set	2	n
OPTION_ICACHE_LIMIT_WIDTH	Maximum address width	32	n

Memory Management Unit (MMU) parameters

Parameter	Description	Default	Values
FEATURE_DMMU	Enable the data bus MMU	NONE	ENABLED
FEATURE_DMMU_HW_TLB_RELOAD	Enable hardware TLB reload	NONE	ENABLED
OPTION_DMMU_SET_WIDTH	Set address width	6	n
OPTION_DMMU_WAYS	Number of ways per set	1	n
FEATURE_IMMU	Enable the instruction bus MMU	NONE	ENABLED
FEATURE_IMMU_HW_TLB_RELOAD	Enable hardware TLB reload	NONE	ENABLED
OPTION_IMMU_SET_WIDTH	Set address width	6	n
OPTION_IMMU_WAYS	Number of ways per set	1	n

System bus parameters

Parameter	Description	Default
FEATURE_STORE_BUFFER	Load store unit store buffer	ENABLED
OPTION_STORE_BUFFER_DEPTH_WIDTH	Load store unit store buffer depth	8
BUS_IF_TYPE	Bus interface type	WISHBONE32
IBUS_WB_TYPE	Instruction bus interface	B3_READ_BURSTING
DBUS_WB_TYPE	Data bus interface type option	CLASSIC

Hardware unit configuration parameters

Parameter	Description	Default
FEATURE_TRACEPORT_EXEC	Traceport hardware interface	NONE
FEATURE_DEBUGUNIT	Hardware breakpoints and debug unit	NONE
FEATURE_PERFCOUNTERS	Performance counters unit	NONE
OPTION_PERFCOUNTERS_NUM	Performance counters to generate	0

Parameter	Description	Default
FEATURE_TIMER	Internal OpenRISC timer	ENABLED
FEATURE_PIC	Internal OpenRISC PIC	ENABLED
OPTION_PIC_TRIGGER	PIC trigger mode	LEVEL
OPTION_PIC_NMI_WIDTH	Non maskable interrupts width	0
OPTION_RF_CLEAR_ON_INIT	clearing all registers on initialization	0
OPTION_RF_NUM_SHADOW_GPR	Number of shadow register files	0
OPTION_RF_ADDR_WIDTH	Address width of the register file	5
OPTION_RF_WORDS	Number of registers in the register file	32
FEATURE_FASTCONTEXTS	Fast context switching of register sets	NONE
FEATURE_MULTICORE	<code>coreid</code> and <code>numcores</code> SPR registers	NONE
FEATURE_FPU	FPU, for cappuccino pipeline only	NONE
OPTION_FTOI_ROUNDING	Rounding behavior for <code>lf.ftoi.s</code>	CPP
FEATURE_BRANCH_PREDICTOR	Branch predictor implementation	SIMPLE

Note: *C/C++ double to integer conversion assumes truncation (rounding toward zero). The default (CPP) value of `OPTION_FTOI_ROUNDING` forces toward zero rounding mode exclusively for `lf.ftoi.s` instruction regardless of rounding mode bits of FPCSR. While with IEEE value `lf.ftoi.s` performs conversion in according with rounding mode bits of FPCSR. And these bits are nearest-even rounding mode by default. All other floating point instructions always perform rounding in according with rounding mode bits of FPCSR.*

Exception handling options

Parameter	Description	Default
FEATURE_DSX	Enable setting the <code>SR[DSX]</code> flag	ENABLED
FEATURE_RANGE	Enable checking and raising range exceptions	ENABLED
FEATURE_OVERFLOW	Enable checking and raising overflow exceptions	ENABLED

ALU configuration options

Parameter	Description	Default
FEATURE_MULTIPLIER	Specify the multiplier implementation	THREESTAGE
FEATURE_DIVIDER	Specify the divider implementation	SERIAL
OPTION_SHIFTER	Specify the shifter implementation	BARREL
FEATURE_CARRY_FLAG	Enable checking and setting the carry flag	ENABLED

Instruction enabling options

Parameter	Description	Default
FEATURE_MAC	1. mac* multiply accumulate instructions	NONE
FEATURE_SYSCALL	1. sys OS syscall instruction	ENABLED
FEATURE_TRAP	1. trap instruction	ENABLED
FEATURE_ADDC	1. addc add with carry flag instruction	ENABLED
FEATURE_SRA	1. sra shift right arithmetic instruction	ENABLED
FEATURE_ROR	1. ror* rotate right instructions	NONE
FEATURE_EXT	1. ext* sign extend instructions	NONE
FEATURE_CMOV	1. cmov conditional move instruction	ENABLED
FEATURE_FFL1	1. f[f1] 1 find first/last set bit instructions	ENABLED
FEATURE_ATOMIC	1. lwa and 1. swa atomic instructions	ENABLED
FEATURE_CUST1	1. cust* custom instruction	NONE
FEATURE_CUST2	1. cust* custom instruction	NONE
FEATURE_CUST3	1. cust* custom instruction	NONE
FEATURE_CUST4	1. cust* custom instruction	NONE
FEATURE_CUST5	1. cust* custom instruction	NONE
FEATURE_CUST6	1. cust* custom instruction	NONE
FEATURE_CUST7	1. cust* custom instruction	NONE
FEATURE_CUST8	1. cust* custom instruction	NONE