Multicore Designer User Guide

The user interface of the application is based on `GTK+ 2.0`. Before running the application, the user should install the `GTK+ 2.0` library.

Installation Instructions for Windows users

NOTE: These instructions have been tested on a 64-bit Intel-based system running Windows 8.0. The instructions may work on other systems, but we have not tested them elsewhere. The instructions are based on: https://www.gtk.org/download/windows.php

Step 1: Install MSYS2

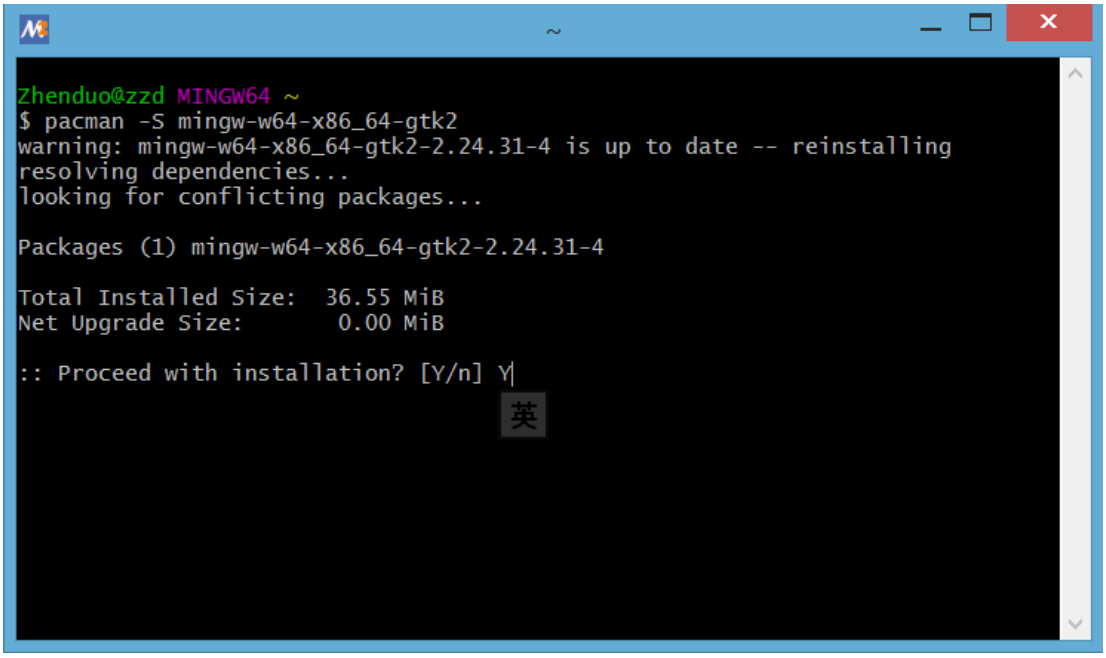
Download the 64-bit `MSYS2` installer (`msys2-x86_64-20161025.exe`) and follow the installation instructions. The link is http://www.msys2.org/

Step 2: Install `GTK+2` and its dependencies

Open a `MSYS2` shell by double-clicking the `MSYS2` icon, and run the command:

```
pacman -S mingw-w64-x86_64-gtk2
```

Answer “`Y`” to proceed with the installation.



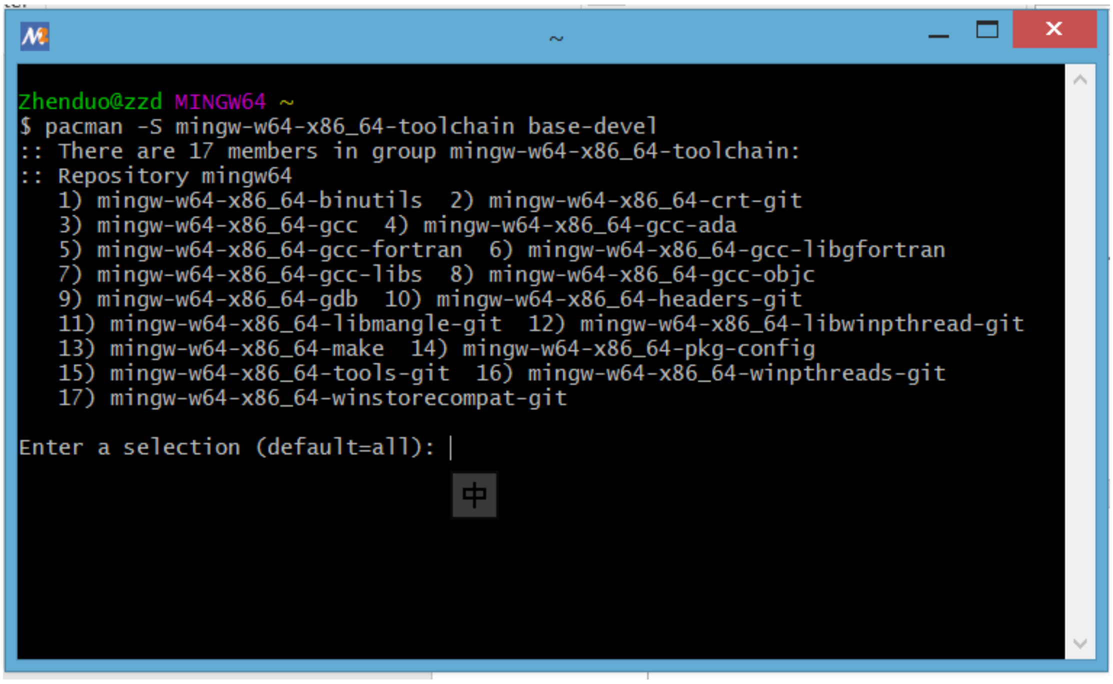
```
Zhenduo@zzd MINGW64 ~
$ pacman -S mingw-w64-x86_64-gtk2
warning: mingw-w64-x86_64-gtk2-2.24.31-4 is up to date -- reinstalling
resolving dependencies...
looking for conflicting packages...

Packages (1) mingw-w64-x86_64-gtk2-2.24.31-4

Total Installed Size:  36.55 MiB
Net Upgrade Size:       0.00 MiB

:: Proceed with installation? [Y/n] Y
```

Step 3: Install the build tools

If you want to develop a `GTK+2` application in other languages like `C`, `C++`, `Fortran`, etc., you'll need a compiler like `gcc` and other development tools:

```
pacman -S mingw-w64-x86_64-toolchain base-devel
```

Respond to the question the terminal asks by simply pressing `ENTER`.



```
Zhenduo@zzd MINGW64 ~
$ pacman -S mingw-w64-x86_64-toolchain base-devel
:: There are 17 members in group mingw-w64-x86_64-toolchain:
:: Repository mingw64
   1) mingw-w64-x86_64-binutils  2) mingw-w64-x86_64-crt-git
   3) mingw-w64-x86_64-gcc  4) mingw-w64-x86_64-gcc-ada
   5) mingw-w64-x86_64-gcc-fortran  6) mingw-w64-x86_64-gcc-libgfortran
   7) mingw-w64-x86_64-gcc-libs  8) mingw-w64-x86_64-gcc-objc
   9) mingw-w64-x86_64-gdb  10) mingw-w64-x86_64-headers-git
   11) mingw-w64-x86_64-libmangle-git  12) mingw-w64-x86_64-libwinpthread-git
   13) mingw-w64-x86_64-make  14) mingw-w64-x86_64-pkg-config
   15) mingw-w64-x86_64-tools-git  16) mingw-w64-x86_64-winpthreads-git
   17) mingw-w64-x86_64-winstorecompat-git

Enter a selection (default=all): |
```

Installation Instructions for Mac users

NOTE: These instructions have been tested on a 64-bit Intel-based system running `OS-X 10.10` (Yosemite), `10.11` (El Capitan), and `macOS 10.15` (Catalina). The instructions may work on other systems, but we have not test them elsewhere yet. The application will **not** work on `OS-X 10.9` (Mavericks) based systems.

Step 1: Install X11

Install `XQuartz X11` for Mac (https://www.xquartz.org/), if it is not already installed.

Step 2: Install Homebrew

First, start a `bash` shell by opening the terminal and issuing the following command:

```
bash
```

Then, issue the following command on the terminal:

```
/usr/bin/ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Step 3: Install `GTK+2` and corresponding libraries

Issue the following two commands on the terminal:

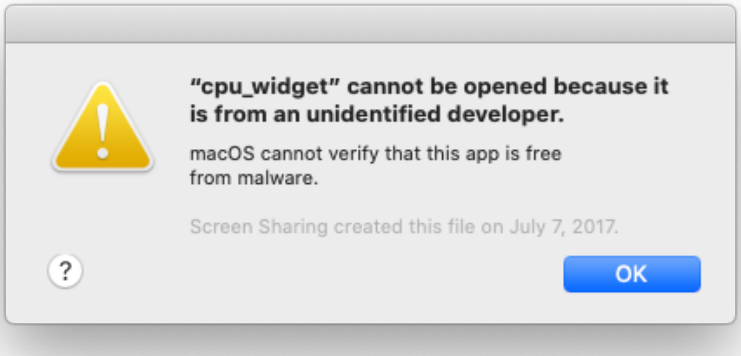
```
brew install gtk+
```

```
pkg-config --libs --cflags gtk+-2.0
```

Step 4: Verify the trustworthiness of the application with `macOS`

Modern versions of the `macOS` include a technology called Gatekeeper, to ensure that only trusted software runs on your Mac. By default, your Mac will refuse to run the application. You will need to explicitly allow it. To do so, follow these steps:

Double-click the file `multicore_designer`. This attempts to run the application. You may get a warning that looks like this:

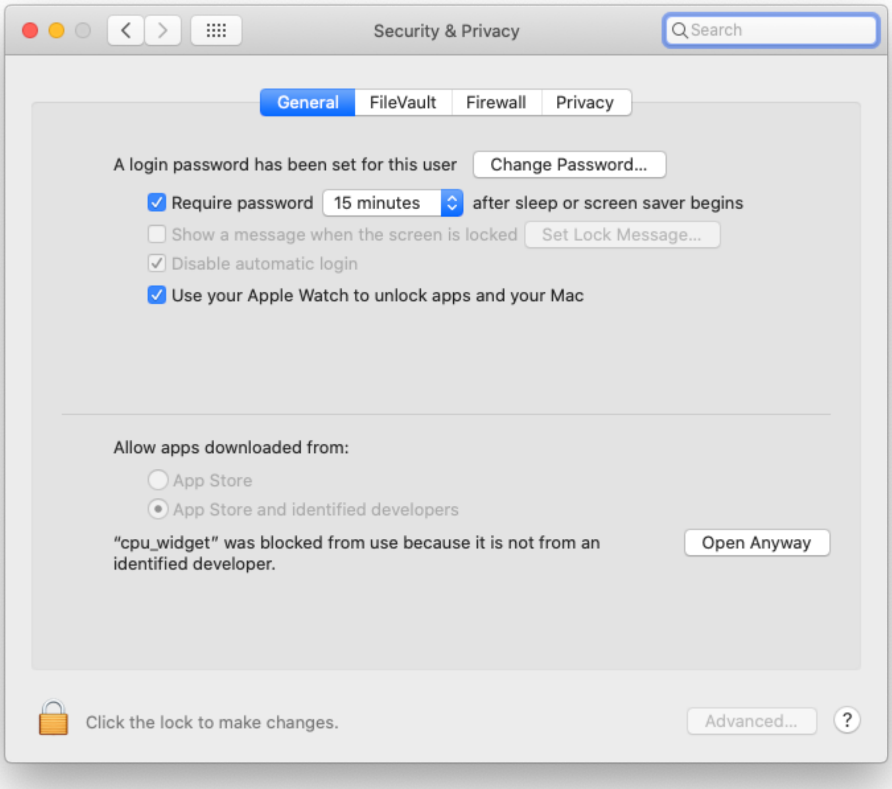


"cpu_widget" cannot be opened because it is from an unidentified developer.

macOS cannot verify that this app is free from malware.

Screen Sharing created this file on July 7, 2017.

Go to the Apple sign on the top left of the screen → System Preferences → Security & Privacy and select "Open Anyway". This will allow the system to run the application.



A login password has been set for this user

☑ Require password 15 minutes after sleep or screen saver begins

☐ Show a message when the screen is locked

☑ Disable automatic login

☑ Use your Apple Watch to unlock apps and your Mac

Allow apps downloaded from:

○ App Store

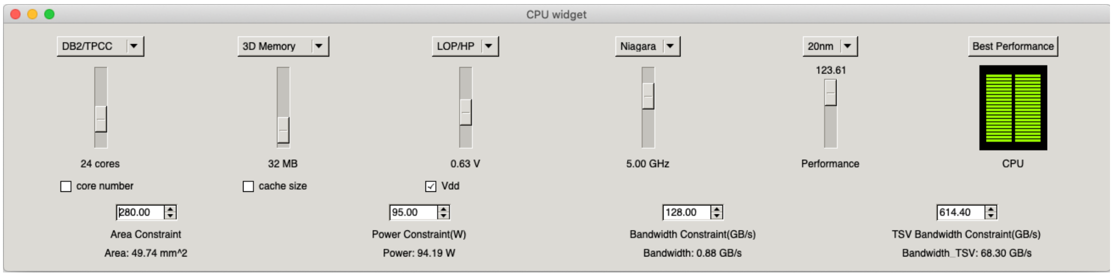
● App Store and identified developers

"cpu_widget" was blocked from use because it is not from an identified developer.

Click the lock to make changes.

Introduction to the User Interface

A picture of the user interface is shown below



24 cores
☐ core number
280.00
Area Constraint
Area: 49.74 mm^2

32 MB
☐ cache size

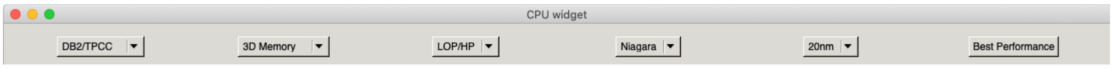
0.63 V
☑ Vdd
95.00
Power Constraint(W)
Power: 94.19 W

5.00 GHz
128.00
Bandwidth Constraint(GB/s)
Bandwidth: 0.88 GB/s

614.40
TSV Bandwidth Constraint(GB/s)
Bandwidth_TSV: 68.30 GB/s

Non-derived Modeling Parameters

The five pull-down lists at the top set the parameters in which the system will be evaluated.



The current options include the following

Application	Memory (Cache) Type	Transistor Type	Core Type	Tech Node
DB2/TPCC	Plain Mem	LOP	Niagara	65 nm
DB2/TPCH	3D Mem	LOP/HP	FPGA	45 nm
Apache/Web		HP	ASIC	32 nm
			ARM	20 nm

Application

The **DB2/TPCC** and **DB2/TPCH** applications are models of the corresponding TPC benchmarks running against a DB2 database. The **Apache/Web** application models a SPECWeb workload running on an Apache web server. Different applications have different cache reuse characteristics, off-chip memory bandwidth demands, working set sizes, and parallelism.

Memory Type

This category refers to on-chip memory, in particular the last-level on-chip cache (LLC), and is not to be confused with off-chip memory technologies like DRAM. **Plain Mem** refers to a conventional 2D-planar LLC, on the same die as the cores. **3D Mem** refers to LLC that is implemented on its own die, which is 3D-stacked on top of the die that implements cores. Communication between the cores and the LLC is achieved using Through-Silicon Vias (TSVs), which are significantly shorter, faster, higher bandwidth, and lower energy than conventional on-chip planar core-cache networks. Overall, more cache (on-chip memory) consumes more power and area, but stores more data on chip and reduces the dependency of the application on high-speed data transfers from off-chip main memory (DRAM).

Transistor Type

LOP models low-operational power transistors following ITRS 2009 specifications for both cores and cache. **LOP/HP** models LOP transistors for the cache gut high-performance transistors for the cores. **HP** models high-performance transistors for both the cores and the cache. The transistor type provides a trade-off between energy consumption and speed: LOP transistors consume the least energy but at a performance loss, while HP are energy-hungry but fast. The transistor type, for a given tech node, defines the range of supply voltage (V_{dd}) and threshold voltage (V_{th}) that will be assumed in the evaluation. In turn, these voltages will determine the maximum clock rate of the processor and its energy consumption: higher voltage allows for linearly faster clock rates, but at quadratically higher power consumption.

Core Type

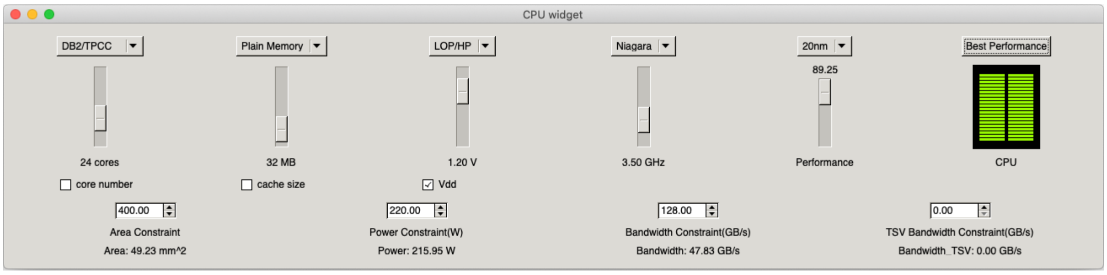
This list defines the core implementation technology that will be assumed by the model. **Niagara** and **ARM** model cores similar to Sun UltraSparc T1 and ARM cores, respectively. **FPGA** models cores with performance, power and area characteristics similar to the ones presented by application-specific functionality implementations on FPGAs. **ASIC** pushes such characteristics to the limit by assuming efficiencies closer to full-custom application-specific circuits. Different core types present different trade-offs in the performance, power, and area occupancy dimensions.

Tech Node

The technology node defines the semiconductor node that will be assumed in the performance and energy evaluation. The parameters of the node affect the performance, area occupancy, energy consumption and supply voltage of the LOP and HP transistors, and the circuits they implement for cores and caches.

Automatically Derived Best Performance Configuration

The `Best Performance` button is used to find the configuration of number of cores (of a given type), cache size (of a given memory type), V_{dd} (restricted by the transistor type and technology node) and frequency (determined by the core type, voltage and tech node) which can give the best speedup while conforming to the constraints of area, power and bandwidth.



DB2/TPCC
Plain Memory
LOP/HP
Niagara
20nm
89.25

24 cores
32 MB
1.20 V
3.50 GHz

☐ core number
☐ cache size
☑ Vdd

400.00 Area Constraint
Area: 49.23 mm^2

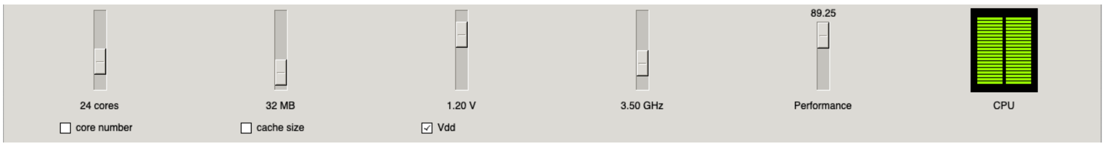
220.00 Power Constraint(W)
Power: 215.95 W

128.00 Bandwidth Constraint(GB/s)
Bandwidth: 47.83 GB/s

0.00 TSV Bandwidth Constraint(GB/s)
Bandwidth_TSV: 0.00 GB/s

For example, with the selection of the parameters above (running the DB2/TPCC workload on a multicore with Niagara-like cores at 3.5 GHz and a 2D-planar cache, implemented with LOP/HP transistors at 20 nm with 1.2 V supply voltage), and under the constraints of $400mm^2$ maximum die area, 220 W peak power and 128 GB/s peak off-chip bandwidth, the maximum performance is achieved by a multicore with 24 cores and 32 MB last-level cache. These parameters, for example, are very similar to a high-end Intel Skylake processor.

Automatically Derived Architectural Parameters



24 cores
☐ core number
32 MB
☐ cache size
1.20 V
☑ Vdd
3.50 GHz
89.25
Performance
CPU

The second row of the user interface consists of several sliders and checkboxes. The sliders correspond to architectural parameters that the application calculates in order to arrive at a configuration with the highest performance, subject to physical constraints.

Core Count

The first slider represents the number of cores. If the user moves the slider, the number of processor cores changes accordingly (max 192). At the same time, the other sliders will also change to achieve the highest performance, under the specified number of cores. The core type is specified

Cache Size

The second slider is used to set the size of the cache (max 512 MB). The

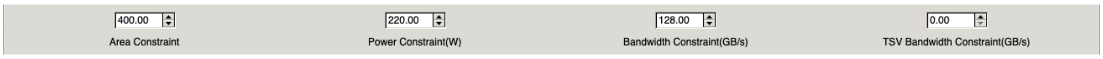
Supply Voltage and Clock Frequency

The third slider is used to set the supply voltage (V_{dd}). The fourth slider is used to set the clock frequency of the processor. If the user moves any one of those four sliders, the other three will also be changed automatically to achieve the highest performance except for the case that the slider is locked by clicking the check box right under it.

Locking Architectural Parameters

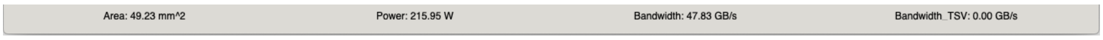
The three check boxes below the sliders are used to lock the corresponding slider. When the check box is checked, the user cannot move the slider above directly or indirectly by moving other sliders. The V_{dd} check box locks both the V_{dd} and the frequency sliders. In addition, when moving the frequency slider, the V_{dd} slider is considered locked. The last slider and the CPU widget is used to show the performance of the processor. Note that when the user moves the core number slider or cache size slider, the performance value does not change linearly. The max number of cores is 192 and the max size of a cache is 512 MB.

Physical Constraints



400.00
Area Constraint
220.00
Power Constraint(W)
128.00
Bandwidth Constraint(GB/s)
0.00
TSV Bandwidth Constraint(GB/s)

The third row is used to set the physical constraints on area, power, off-chip bandwidth (to main memory) and TSV bandwidth (the bandwidth between cores and the LLC when using a 3D cache design; applicable only when the 3D Mem type is selected). The default value for each constraint is set automatically according to the data sheet, but the user can also set them to specific values. The Best Performance calculation is constrained by these values.



Area: 49.23 mm^2
Power: 215.95 W
Bandwidth: 47.83 GB/s
Bandwidth_TSV: 0.00 GB/s

The last row of the user interface shows the estimated area occupancy, power consumption, off-chip (to main memory) bandwidth consumption, and TCV bandwidth consumption (for 3D caches). These results are calculated for the design with the architectural parameters indicated by the sliders and the non-derived modeling parameters indicated by the pull-down lists.

More details on the model, the workload configurations, and design parameters can be found at the IEEE Micro 2010 paper [2] and Hardavellas' 2009 PhD Thesis in CS at CMU [1]. The model codified in this tool has been used to forecast the advent of dark silicon and the promise of specialized application-specific accelerators [1, 2], as well as to derive the designs of Scale Out Processors [3], which formed the basis of Thunder-X, Cavium's first server-class ARMv8 processor [4].

References

[1] N. Hardavellas. *Chip-Multiprocessors for Server Workloads*. Ph.D. Thesis, Department of Computer Science, Carnegie Mellon University, Pittsburgh, 2009.

[2] N. Hardavellas, M. Ferdman, B. Falsafi and A. Ailamaki. *Toward Dark Silicon in Servers*. In IEEE Micro, Special Issue on Big Chips, Vol. 31(4), pp. 6-15, July/August 2011.

[3] P. Lotfi-Kamran, B. Grot, M. Ferdman, S. Volos, Y. O. Koçberber, J. Picorel, A. Adileh, D. Jevdjic, S. Idgunji, E. Özer, and B. Falsafi. *Scale-Out Processors*. in Proceedings of the 39th International Symposium on Computer Architecture (ISCA), 2012.

[4] J. De Gelas. *Investigating Cavium's ThunderX: The First ARM Server SoC With Ambition*. https://www.anandtech.com/show/10353/investigating-cavium-thunderx-48-arm-cores, 2016.