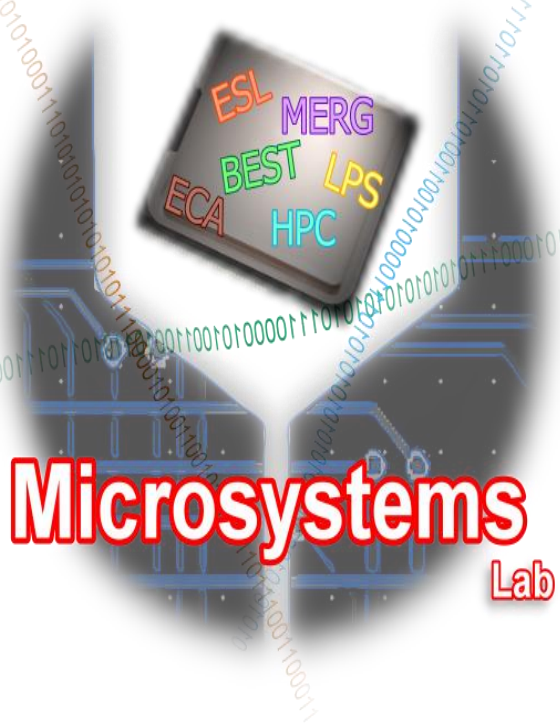
Get familiar with Simplescalar and Wattch



Advanced Computer Architecture

Lab1

Use with putty.exe

•IP Address : 140.118.115.163

•Username and password are both your ID

Ex: d10602805

TA: D10602805@mail.ntust.edu.tw

Walle Haileeyesus E.

2024/12/10

Outline

1. Introduction
2. Simplescalar
3. Wattch
4. Installation
5. Hint

1. Introduction

- Task #1 Install and Understand Simplescalar
- Task #2 Install and understand Wattch
- Task #3 Finish the table with these tools and Explain your findings (Evaluate performance, power/energy consumption)

2. SimpleScalar

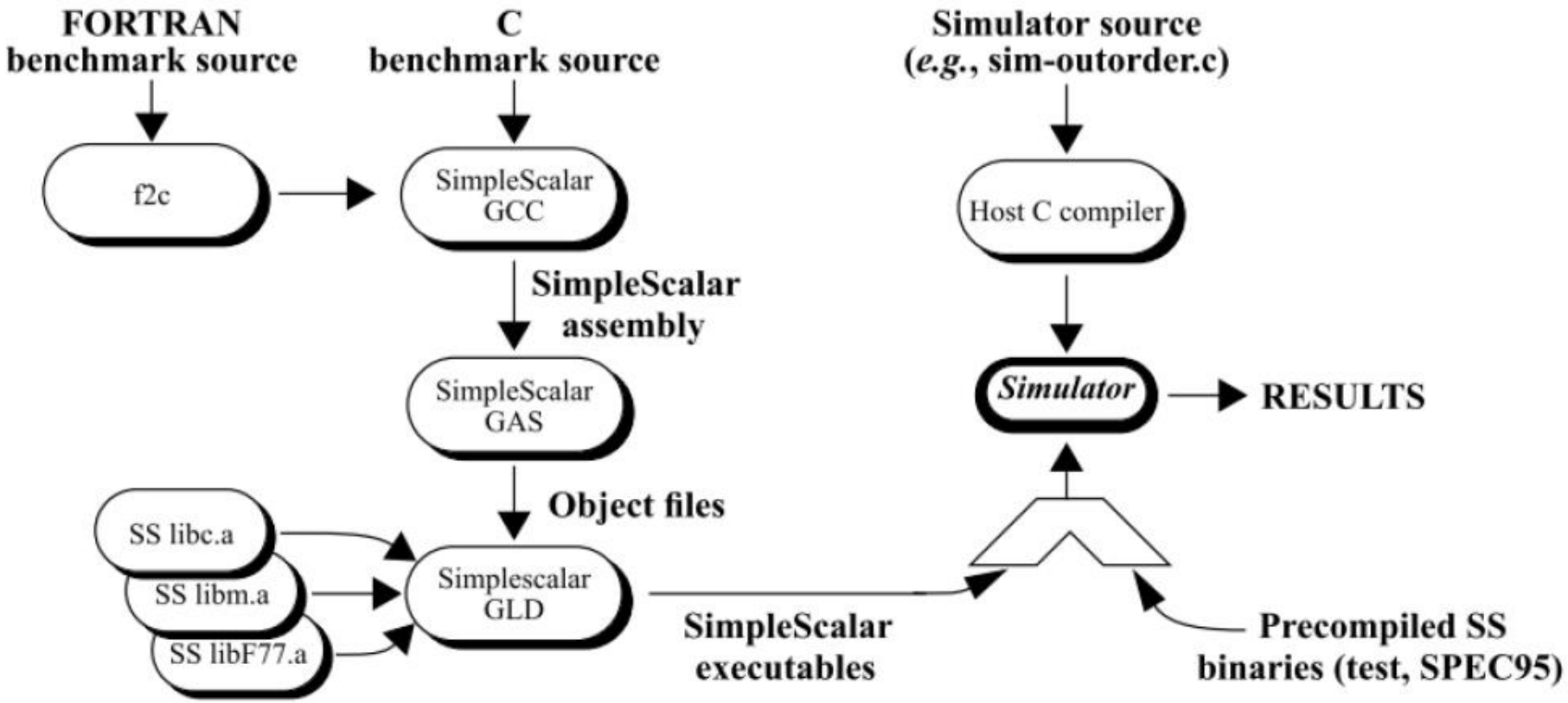
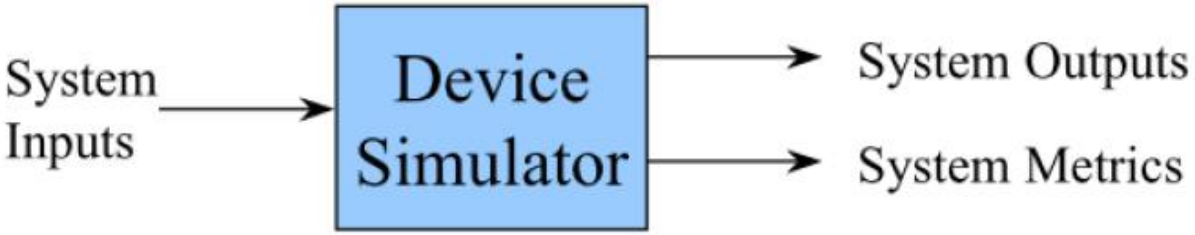


Figure 1. SimpleScalar tool set overview

2. SimpleScalar

- What is an architectural simulator?
  - tool that reproduces the behavior of a computing device

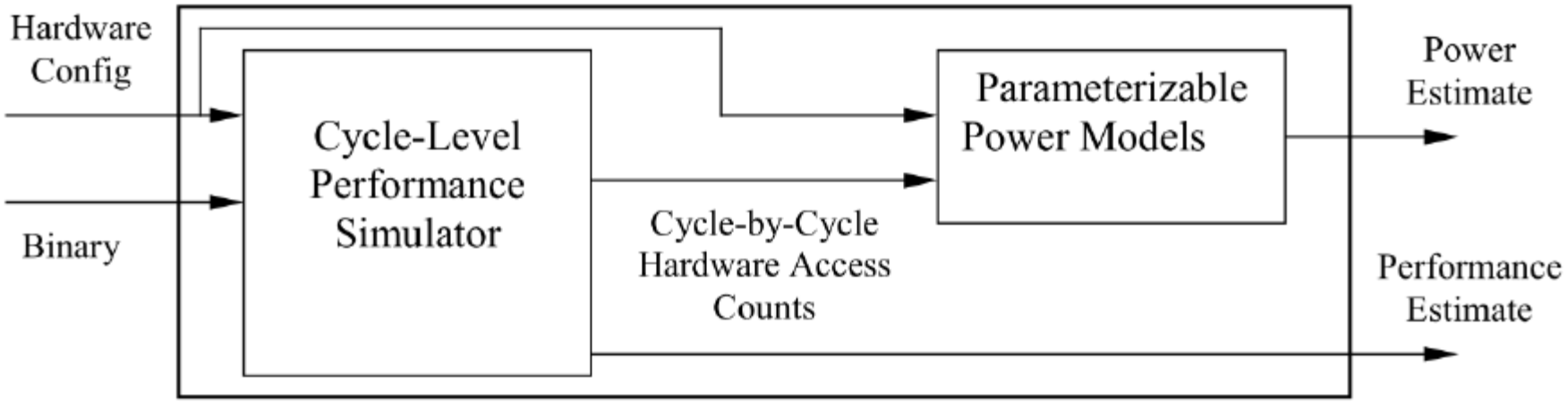


- Why use a simulator?
  - leverage faster, more flexible S/W development cycle
    - permits more design space exploration
    - facilitates validation before H/W becomes available
    - level of abstraction can be throttled to design task
    - possible to increase/improve system instrumentation

2. SimpleScalar

```
sim: ** simulation statistics **
sim_num_insn                 27335 # total number of instructions executed
sim_num_refs                  9316 # total number of loads and stores executed
sim_elapsed_time                 1 # total simulation time in seconds
sim_inst_rate            27335.0000 # simulation speed (in insts/sec)

sim_inst_class_prof    # instruction class profile
sim_inst_class_prof.array_size = 7
sim_inst_class_prof.bucket_size = 1
sim_inst_class_prof.count = 7
sim_inst_class_prof.total = 27334
sim_inst_class_prof.imin = 0
sim_inst_class_prof.imax = 7
sim_inst_class_prof.average = 3904.8571
sim_inst_class_prof.std_dev = 4492.1382
sim_inst_class_prof.overflows = 0
# pdf == prob dist fn, cdf == cumulative dist fn
#          index      count    pdf
sim_inst_class_prof.start_dist
load                    4313  15.78
store                   5003  18.30
uncond branch            751   2.75
cond branch             4444  16.26
int computation        12801  46.83
```

3. Wattch



Hardware Structure	Model Type
Instruction Cache	Cache Array (2x bitlines)
Wakeup Logic	CAM
Issue Selection Logic	Complex combinational
Instruction window	Array/CAM
Branch Predictor	Cache Array (2x bitlines)
Register File	Array (1x bitlines)
Translation Lookaside Buffer	Array/CAM
Load/Store Queue	Array/CAM
Data Cache	Cache Array (2x bitlines)
Integer Functional Units	Complex combinational
FP Functional Units	Complex combinational
Global Clock	Clock

3. Wattch

Four categories of processor units

$$P_d = CV_{dd}^2 af$$

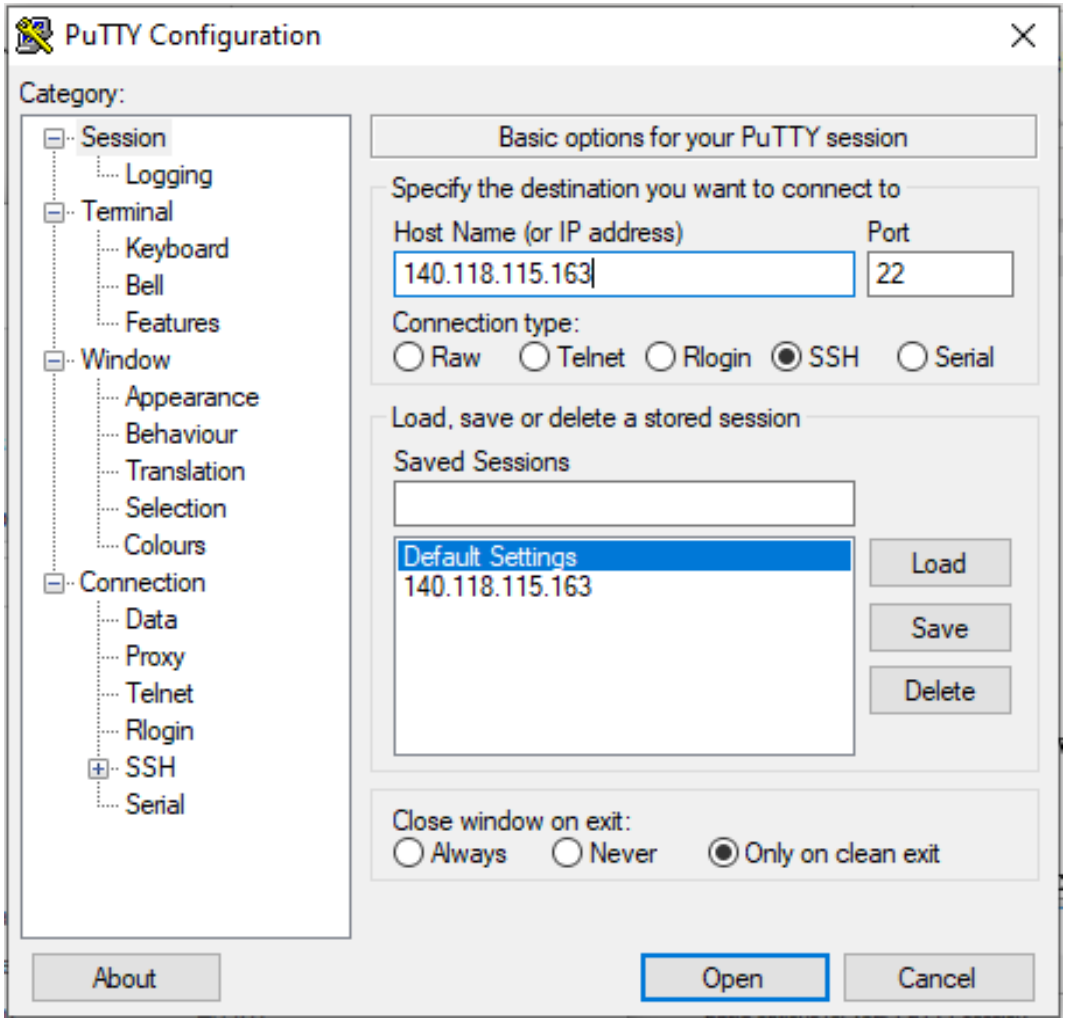
Process technology

model

Collected from simulator or based on certain assumptions

- Array structures: data and instruction caches, register files, …
- Fully associative content-addressable memories: TLBs, …
- Combinational logic and wires: functional units, …
- Clocking: clock buffers, clock wires, …

4. Installation



4. Installation(Simpleutils)

```
d10602805@lab701:~$ cd lab1/
d10602805@lab701:~/lab1$ ls
benchmark  simplescalar
d10602805@lab701:~/lab1$ cd simplescalar/
d10602805@lab701:~/lab1/simplescalar$ ls
ar                      ranlib                  simpleutils-990811.tar.gz
flex-2.5.4a.tar.gz      simplesim-3v0d.tgz      sim-wattch-1.02d.tar.gz
gcc-2.7.2.3.ss.tar.gz   simpletools-2v0.tgz
d10602805@lab701:~/lab1/simplescalar$
```

```
d10602805@lab701:~/lab1/simplescalar$ tar xzvf simpleutils-990811.tar.gz
```

```
d10602805@lab701:~/lab1/simplescalar$ cd simpleutils-990811
d10602805@lab701:~/lab1/simplescalar/simpleutils-990811$ ls
bfd             configure      include       makefile.vms     mpw-install
binutils        configure.in   install-sh    missing          mpw-README
ChangeLog       COPYING        intl          mkdep            opcodes
config          COPYING.LIB    ld            mkinstalldirs    README
config.guess    CVS            libiberty     move-if-change   setup.com
config.if       etc            ltconfig      mpw-build.in     symlink-tree
config-ml.in    gas            ltmain.sh     mpw-config.in    texinfo
config.sub      gprof          Makefile.in   mpw-configure    ylwrap
d10602805@lab701:~/lab1/simplescalar/simpleutils-990811$
```

4. Installation(Simpleutils)

```
d10602805@lab701:~/lab1/simplescalar/simpleutils-990811$ cd ld/
d10602805@lab701:~/lab1/simplescalar/simpleutils-990811/ld$ ls
acinclude.m4     emultempl       ldfile.h        ldver.c         mpw-make.sed
aclocal.m4       gen-doc.texi    ldgram.y        ldver.h         mri.c
ChangeLog        genscripts.sh   ld.h            ldwrite.c       mri.h
config.in        h8-doc.texi     ldint.texinfo   ldwrite.h       NEWS
configure        ld.1            ldlang.c        lexsup.c        pe-dll.c
configure.host   ldcref.c        ldlang.h        mac-ld.r        po
configure.in     ldctor.c        ldlex.h         Makefile.am     README
configure.tgt    ldctor.h        ldlex.l         Makefile.in     scripttempl
CVS              ldemul.c        ldmain.c        mpw-config.in   stamp-h.in
deffile.h        ldemul.h        ldmain.h        mpw-elfmips.c   sysdep.h
deffilep.y       ldexp.c         ldmisc.c        mpw-eppcmac.c   testsuite
dep-in.sed       ldexp.h         ldmisc.h        mpw-esh.c       TODO
emulparams       ldfile.c        ld.texinfo      mpw-idtmips.c
d10602805@lab701:~/lab1/simplescalar/simpleutils-990811/ld$ vim ldlex.l
```

Use the search finction(:?yy_cu) to find yy_current_buffer and replace that with YY_CURRENT_BUFFER

```
tatic void
y_input (buf, result, max_size)
    char *buf;
    int *result;
    int max_size;

 *result = 0;
 if (yy_current_buffer->yy_input_file)
   {
     if (yyin)
       {
         *result = read (fileno (yyin), (char *) buf, max_size);
         if (*result < 0)
           einfo ("%F%P: read in flex scanner failed\n");
       }
```

4. Installation(Simpleutils)

```
d10602805@ACA-Server:~/lab1/simplescalar/simpleutils-990811/ld$ cd ..
d10602805@ACA-Server:~/lab1/simplescalar/simpleutils-990811$ ./configure  --targ
et=sslittle-na-sstrix --with-gnu-as --with-gnu-ld --prefix=/home/d10602805/lab1/
simplescalar
```

./configure --target=sslittle-na-sstrix --with-gnu-as --with-gnu-ld --prefix=/home/<yourID>/lab1/simplescalar

```
loading cache ../config.cache
checking for a BSD compatible install... (cached) /usr/bin/install -c
creating ./config.status
creating Makefile
d10602805@lab701:~/lab1/simplescalar/simpleutils-990811$ make  CFLAGS=-O
```

** CFLAGS=-O (Big O)

```
make[3]: Leaving directory '/home/d10602805/lab1/simplescalar/simpleutils-990811
/ld'
make[2]: Leaving directory '/home/d10602805/lab1/simplescalar/simpleutils-990811
/ld'
make[1]: Leaving directory '/home/d10602805/lab1/simplescalar/simpleutils-990811
/ld'
d10602805@lab701:~/lab1/simplescalar/simpleutils-990811$ make install
```

4. Installation(Simplesim)

```
henry@mergubuntu:~/lab1/simplescalar/simpleutils-990811$ cd ..
henry@mergubuntu:~/lab1/simplescalar$ ls
ar                  gcc-2.7.2.3.ss.tar.gz  man     simplesim-3v0d.tgz   simpleutils-990811.tar.gz
bin                 include                ranlib  simpletools-2v0.tgz  sim-wattch-1.02d.tar.gz
flex-2.5.4a.tar.gz  lib                    share   simpleutils-990811   sslittle-na-sstrix
henry@mergubuntu:~/lab1/simplescalar$ tar zxvf simplesim-3v0d.tgz
```

```
henry@mergubuntu:~/lab1/simplescalar$ cd simplesim-3.0/
henry@mergubuntu:~/lab1/simplescalar/simplesim-3.0$ ls
ANNOUNCE-3.0  dlite.h    hack_guide.pdf  main.c       ptrace.c            redir.sh     sim-outorder.c  target-alpha
bitmap.h      eio.c      hack_guide.ps   Makefile     ptrace.h            regs.c       sim-profile.c   target-pisa
bpred.c       eio.h      host.h          memory.c     range.c             regs.h       sim-safe.c      tests
bpred.h       endian.c   libexo          memory.h     range.h             resource.c   stats.c         tests-alpha
cache.c       endian.h   LICENSE         misc.c       README              resource.h   stats.h         tests-pisa
cache.h       eval.c     loader.c        misc.h       README.eio          sim-bpred.c  symbol.c        textprof.pl
config        eval.h     loader.h        options.c    README.retarget     sim-cache.c  symbol.h        version.h
config.h      eventq.c   machine.c       options.h    README.sim-inorder  sim-eio.c    syscall.c       WARRANTY
CONTRIBUTORS  eventq.h   machine.def     pipeview.pl  README.windows      sim-fast.c   syscall.h
dlite.c       FAQ        machine.h       PROJECTS     redir.bash          sim.h        sysprobe.c
henry@mergubuntu:~/lab1/simplescalar/simplesim-3.0$ make config-pisa
rm -f config.h machine.h machine.c machine.def loader.c symbol.c syscall.c
ln -s target-pisa/config.h config.h
ln -s target-pisa/pisa.h machine.h
ln -s target-pisa/pisa.c machine.c
ln -s target-pisa/pisa.def machine.def
ln -s target-pisa/loader.c loader.c
ln -s target-pisa/symbol.c symbol.c
ln -s target-pisa/syscall.c syscall.c
rm -f tests
ln -s tests-pisa tests
henry@mergubuntu:~/lab1/simplescalar/simplesim-3.0$ make
```

4. Installation(Simpletools)

```
xo/libexo.a `./sysprobe -libs` -lm
my work is done here...
henry@mergubuntu:~/lab1/simplescalar/simplesim-3.0$ cd ..
henry@mergubuntu:~/lab1/simplescalar$ ls
ar                  gcc-2.7.2.3.ss.tar.gz  man     simplesim-3.0        simpleutils-990811         sslittle-na-sstrix
bin                 include                ranlib  simplesim-3v0d.tgz   simpleutils-990811.tar.gz
flex-2.5.4a.tar.gz  lib                    share   simpletools-2v0.tgz  sim-wattch-1.02d.tar.gz
henry@mergubuntu:~/lab1/simplescalar$ tar zxvf simpletools-2v0.tgz
```

4. Installation(gcc-2.7.2.3)

```
henry@ubuntu:~/lab1/simplescalar$ ls
bin                     include              simpletools-2v0.tgz
f2c-1994.09.27          info                 simpleutils-990811
flex-2.5.4              lib                  simpleutils-990811.tar.gz
flex-2.5.4a.tar.gz      man                  sim-wattch-1.02d.tar.gz
gcc-2.6.3               Readme.gcc-2.7.2.3   ssbig-na-sstrix
gcc-2.7.2.3             share                sslittle-na-sstrix
gcc-2.7.2.3.ss.tar.gz   simplesim-3.0
glibc-1.09              simplesim-3v0d.tgz
henry@ubuntu:~/lab1/simplescalar$ tar zxvf gcc-2.7.2.3.ss.tar.gz
```

```
@mergubuntu:~/lab1/simplescalar$ cd gcc-2.7.2.3/
@mergubuntu:~/lab1/simplescalar/gcc-2.7.2.3$ chmod -R +w .
```

Don’t forget the “.” after +w

4. Installation(gcc-2.7.2.3)

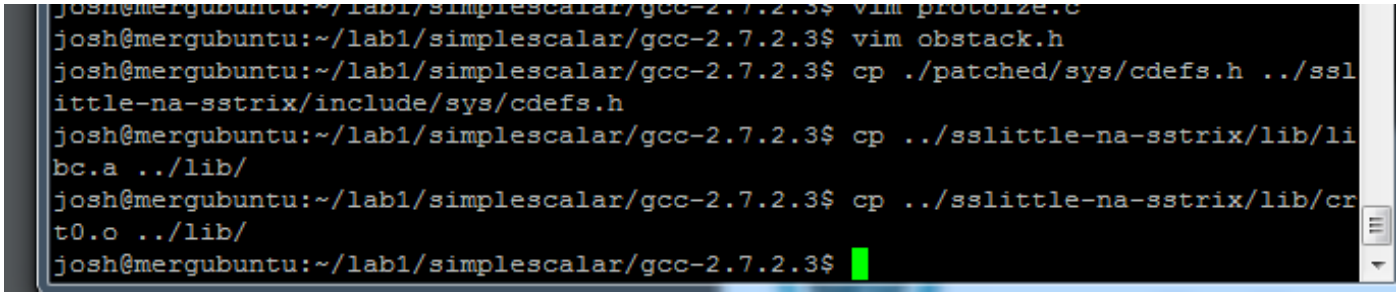
```
   ask for definition of all POSIX facilities.  */
#undef _POSIX_SOURCE
#define _POSIX_SOURCE
#endif

#include <stdarg.h>
/* On some systems stdio.h includes stdarg.h;
   we must bring in varargs.h first.  */
#include <stdio.h>
#include <ctype.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/stat.h>
#ifndef _WIN32
```

In line#60 of file protoize.c, replace #include <varargs.h> with #include<stdarg.h>
Use search function(:60) to go to line 60
** remember to type :wq! after you edit the file!!

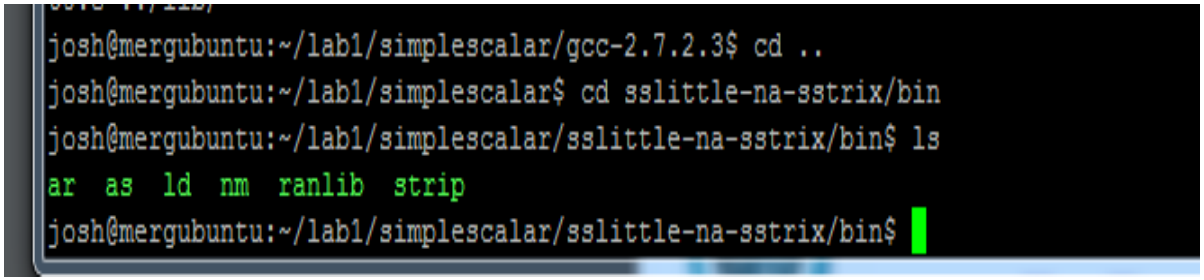
```
#define obstack_ptr_grow(OBSTACK,datum)                         \
__extension__                                                   \
({ struct obstack *__o = (OBSTACK);                             \
   if (__o->next_free + sizeof (void *) > __o->chunk_limit)     \
     _obstack_newchunk (__o, sizeof (void *));                  \
   if (!__o->alloc_failed)                                      \
     *((void **)__o->next_free++) = ((void *)datum);            \
   (void) 0; })

#define obstack_int_grow(OBSTACK,datum)                         \
__extension__                                                   \
({ struct obstack *__o = (OBSTACK);                             \
   if (__o->next_free + sizeof (int) > __o->chunk_limit)        \
```

In line#341 of file obstack.h, replace:
*((void **)__o->next_free)++ = ((void *)datum);\
with
*((void **)__o->next_free++)=((void *)datum;\

4. Installation(gcc-2.7.2.3)



cp ./patched/sys/cdefs.h ../sslittle-na-sstrix/include/sys/cdefs.h
cp ../sslittle-na-sstrix/lib/libc.a ../lib/
cp ../sslittle-na-sstrix/lib/crt0.o ../lib/



4. Installation(gcc-2.7.2.3)

```
josh@mergubuntu:~/lab1/simplescalar$ cd sslittle-na-sstrix/bin
josh@mergubuntu:~/lab1/simplescalar/sslittle-na-sstrix/bin$ ls
ar  as  ld  nm  ranlib  strip
josh@mergubuntu:~/lab1/simplescalar/sslittle-na-sstrix/bin$ rm ar
josh@mergubuntu:~/lab1/simplescalar/sslittle-na-sstrix/bin$ rm ranlib
josh@mergubuntu:~/lab1/simplescalar/sslittle-na-sstrix/bin$ ls
as  ld  nm  strip
josh@mergubuntu:~/lab1/simplescalar/sslittle-na-sstrix/bin$
```

Please make sure you are removing the right file !!!

```
as  ld  nm  strip
josh@mergubuntu:~/lab1/simplescalar/sslittle-na-sstrix/bin$ cp ../../ar ./
josh@mergubuntu:~/lab1/simplescalar/sslittle-na-sstrix/bin$ cp ../../ranlib ./
josh@mergubuntu:~/lab1/simplescalar/sslittle-na-sstrix/bin$ ls
ar  as  ld  nm  ranlib  strip
josh@mergubuntu:~/lab1/simplescalar/sslittle-na-sstrix/bin$ chmod +x ar
josh@mergubuntu:~/lab1/simplescalar/sslittle-na-sstrix/bin$ chmod +x ranlib
josh@mergubuntu:~/lab1/simplescalar/sslittle-na-sstrix/bin$ ls
ar  as  ld  nm  ranlib  strip
josh@mergubuntu:~/lab1/simplescalar/sslittle-na-sstrix/bin$
```

4. Installation(gcc-2.7.2.3)

```
dl0602805@ACA-Server:~/lab1/simplescalar/sslittle-na-sstrix/bin$ cd ../../
dl0602805@ACA-Server:~/lab1/simplescalar$ cd gcc-2.7.2.3/
dl0602805@ACA-Server:~/lab1/simplescalar/gcc-2.7.2.3$ ./configure --target=sslittl
e-na-sstrix --with-gnu-as --with-gnu-ld --prefix=/home/dl0602805/lab1/simplescalar
This appears to be a i686-unknown-linux system.
Using `./config/ss/ss.c' to output insns.
Using `./config/ss/ss.md' as machine description file.
Using `./config/ss/sslittle.h' as target machine macro file.
Using `./config/i386/xm-linux.h' as host machine macro file.
Merged x-linux.
Merged ss/t-ss-gas.
Merged c++ fragment(s).
Created `./Makefile'.
Merged x-linux.
Merged ss/t-ss-gas.
Created `cp/Makefile'.
Links are now set up to build a cross-compiler for sslittle-na-sstrix
  from i686-unknown-linux.
dl0602805@ACA-Server:~/lab1/simplescalar/gcc-2.7.2.3$
```

Go back to the gcc-2.7.2.3 folder and use command:
./configure --target=sslittle-na-sstrix --with-gnu-as --with-gnu-ld --prefix=/home/<yourID>/lab1/simplescalar

4. Installation(gcc-2.7.2.3)

```
Links are now set up to build a cross-compiler for sslittle-na-sstrix
  from i686-unknown-linux.
josh@mergubuntu:~/lab1/simplescalar/gcc-2.7.2.3$ make LANGUAGES=c CFLAGS=-O CC="
gcc -m32"
```

```
insn-output.c:824:5: error: stray ?   ? ? in program
insn-output.c:824:5: error: stray ?   ? ? in program
insn-output.c:824:5: error: stray ?   ? ? in program
insn-output.c:824:5: error: stray ?   ? ? in program
insn-output.c:824:5: error: stray ?   ? ? in program
insn-output.c:824:5: error: stray ?   ? ? in program
insn-output.c:824:5: error: stray ?   ? ? in program
insn-output.c:839:3: warning: missing terminating " character [enabled by defaul
t]
insn-output.c:824:5: error: missing terminating " character
make: *** [insn-output.o] Error 1
josh@mergubuntu:~/lab1/simplescalar/gcc-2.7.2.3$
```

Changed in "insn-output.c": "return "FIXME\n" to "return "FIXME\n\"
(line No. 675,750 and 823)

```
  dslots_jump_filled += 2;

  return "FIXME\n\
sll\t%3,%2,26\n\
\tbgez\t%3,1f\n\
\tsll\t%M0,%L1,%2\n\
\t%(b\t3f\n\
\tmove\t%L0,%z4%)\n\
\n\
```

4. Installation(gcc-2.7.2.3)

```
Links are now set up to build a cross-compiler for sslittle-na-sstrix
  from i686-unknown-linux.
josh@mergubuntu:~/lab1/simplescalar/gcc-2.7.2.3$ make LANGUAGES=c CFLAGS=-O CC="
gcc -m32"
```

```
cxxmain.c: In function ?   alh? ?
cxxmain.c:2885:6: warning: incompatible implicit declaration of built-in functio
n ?   xit? ? [enabled by default]
cxxmain.c:2954:5: warning: incompatible implicit declaration of built-in functio
n ?   ree? ? [enabled by default]
cxxmain.c:2967:3: warning: incompatible implicit declaration of built-in functio
n ?   xit? ? [enabled by default]
cxxmain.c: In function ?   atal? ?
cxxmain.c:2975:3: warning: incompatible implicit declaration of built-in functio
n ?   xit? ? [enabled by default]
cxxmain.c: At top level:
cxxmain.c:2978:8: error: conflicting types for ?   alloc? ?
cxxmain.c:2979:8: warning: conflicting types for built-in function ?   ealloc? ?
 [enabled by default]
make: *** [cxxmain.o] Error 1
josh@mergubuntu:~/lab1/simplescalar/gcc-2.7.2.3$
```

Comment out lines 2978-2979 in file cxxmain.c.

```
  exit (1);
}

//char * malloc ();
//char * realloc ();

char *
xmalloc (size)
     unsigned size;
```

4. Installation(gcc-2.7.2.3)

```
Links are now set up to build a cross-compiler for sslittle-na-sstrix
  from i686-unknown-linux.
josh@mergubuntu:~/lab1/simplescalar/gcc-2.7.2.3$ make LANGUAGES=c CFLAGS=-O CC="
gcc -m32"
```

```
henry@mergubuntu:~/lab1/simplescalar$ cd gcc-2.7.2.3/
henry@mergubuntu:~/lab1/simplescalar/gcc-2.7.2.3$ make install LANGUAGES=c CFLAG
S=-O CC="gcc -m32"
```

```
underscore.c is unchanged
rm -f tmp-dum.c tmp-dum.s
touch stamp-under
henry@ubuntu:~/lab1/simplescalar/gcc-2.7.2.3$ cd ..
henry@ubuntu:~/lab1/simplescalar$ ls
bin                 gcc-2.7.2.3.ss.tar.gz  Readme.gcc-2.7.2.3   simpleutils-990811.tar.gz
f2c-1994.09.27      glibc-1.09             share                sim-wattch-1.02d.tar.gz
flex-2.5.4          include                simplesim-3.0        ssbig-na-sstrix
flex-2.5.4a.tar.gz  info                   simplesim-3v0d.tgz   sslittle-na-sstrix
gcc-2.6.3           lib                    simpletools-2v0.tgz
gcc-2.7.2.3         man                    simpleutils-990811
henry@ubuntu:~/lab1/simplescalar$ cd bin
henry@ubuntu:~/lab1/simplescalar/bin$ ls
sslittle-na-sstrix-addr2line  sslittle-na-sstrix-gasp     sslittle-na-sstrix-ranlib
sslittle-na-sstrix-ar         sslittle-na-sstrix-gcc      sslittle-na-sstrix-readelf
sslittle-na-sstrix-as         sslittle-na-sstrix-ld       sslittle-na-sstrix-size
sslittle-na-sstrix-c++        sslittle-na-sstrix-nm       sslittle-na-sstrix-strings
sslittle-na-sstrix-c++filt    sslittle-na-sstrix-objcopy  sslittle-na-sstrix-strip
sslittle-na-sstrix-g++        sslittle-na-sstrix-objdump
henry@ubuntu:~/lab1/simplescalar/bin$
```

If you can find “sslittle-na-sstrix-gcc”, then you success !

4. Installation(Wattch)

Do the same thing but change to wattch from simplesim-3.0

How to install sim-wattch-1.02d:

1. tar zxvf sim-wattch-1.02d.tar.gz
2. cd sim-wattch-1.02d
3. make config-pisa
4. make

5. Hint(Take susan for example)

After you finished the installation , you may have a folder named “automotive” in benchmark .

```
[henry@ip17-120 ~]$ cd benchmark
[henry@ip17-120 benchmark]$ ls
automotive          consumer.tar.gz  network.tar.gz  security.tar.gz
automotive.tar.gz   network          office.tar.gz   telecomm.tar.gz
[henry@ip17-120 benchmark]$
```

You may have to use the following command to see “automotive” folder as above:

```
$ tar -xf automotive.tar.gz
```

Then go to the folder where has a file named “susan.c” .

```
[henry@ip17-120 benchmark]$ cd automotive
[henry@ip17-120 automotive]$ cd susan
[henry@ip17-120 susan]$ ls
COMPILE  Makefile         input_small.pgm  runme_small.sh  susan.c
LICENSE  input_large.pgm  runme_large.sh   susan
```

5. Hint(Take susan for example)

Edit the Makefile and change that as following(type your ID instead henry):

```
henry@ubuntu:~/lab1/benchmark/automotive/susan$ vim Makefile
henry@ubuntu:~/lab1/benchmark/automotive/susan$ make
```

henry@ubuntu: ~/lab1/benchmark/automotive/susan

```
CC=/home/henry/lab1/simplescalar/bin/sslittle-na-sstrix-gcc

susan: susan.c Makefile
	${CC} -static -O4 -o susan susan.c

clean:
	rm -rf susan output*
~
```

Make this Makefile and try
“/home/yourID/lab1/simplescalar/simplesim-3.0/sim-bpred susan”

5. Hint(Table 1)

You need to fill Table 1 with all available benchmark programs versus **instruction class profiles**.

So you need to reference the file, **The SimpleScalar Tool Set.pdf**, and use the command in section 4.3.

Then you can use the command to get the information:
/home/yourID/lab1/simplescalar/simplesim-3.0/sim-profile -iclass susan

5. Hint(Table 2)

“**Use the detailed simulator (sim-outorder) to measure and describe how the prediction rate affects the processor CPI for your benchmark. Also use this simulator to measure CPI when using the perfect branch predictor type.**”

So you need to reference the file, **The SimpleScalar Tool Set.pdf**, and use the command in page 9.

The default is a bimodal predictor with 2048 entries.”

Then you can use the command to get the information:
/home/yourID/lab1/simplescalar/simplesim-3.0/sim-outorder -bpred taken susan

5. Hint(Table 3)

Have to read these parts:
-decode:width

-issue:width

-issue:inorder

Can use the command like
/home/yourID/lab1/simplescalar/simplesim-3.0/sim-outorder -fetch:ifqsize 8 -decode:width 8 -issue:width 8 -issue:inorder true susan

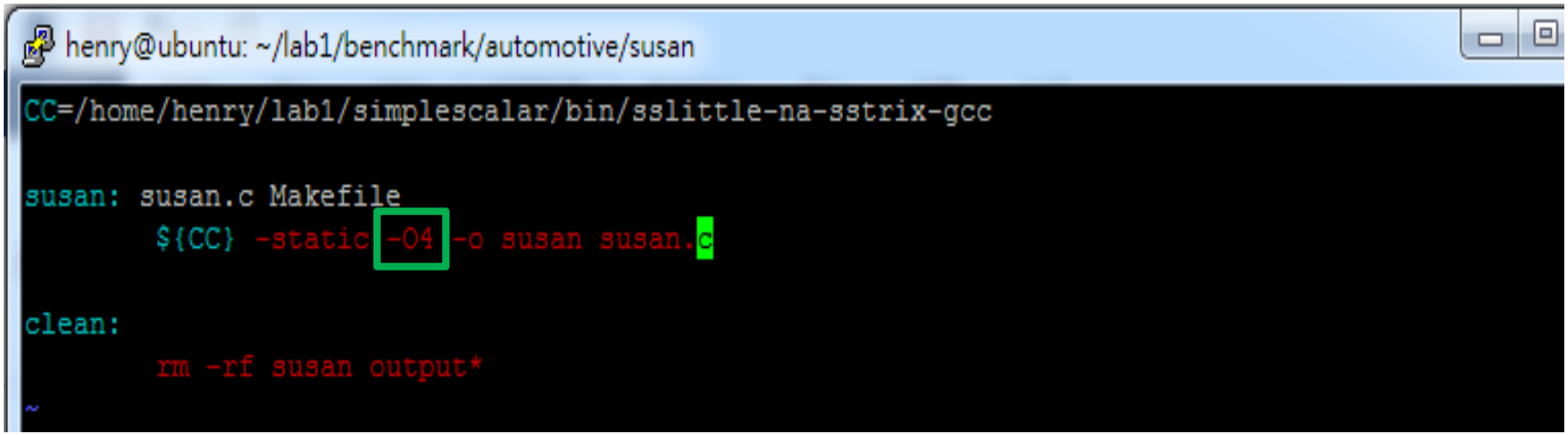
**in-order:true
**out-order: false

5. Hint(Table 4)

Can use the command like
/home/yourID/lab1/simplescalar/sim-wattch-1.02d/sim-outorder susan

**Branch Power:bpred_power

5. Hint(Table 5)



Change –O4 to –O1 and –O2

5. Hint

Sim-Outorder: Detailed Performance Simulator

- generates timing statistics for a detailed out-of-order issue processor core with two-level cache memory hierarchy and main memory
- extra options

`-fetch:ifqsize <size>`	- instruction fetch queue size (in insts)
`-fetch:mplat <cycles>`	- extra branch mis-prediction latency (cycles)
`-bpred <type>`	- specify the branch predictor
`-decode:width <insts>`	- decoder bandwidth (insts/cycle)
`-issue:width <insts>`	- RUU issue bandwidth (insts/cycle)
`-issue:inorder`	- constrain instruction issue to program order
`-issue:wrongpath`	- permit instruction issue after mis-speculation
`-ruu:size <insts>`	- capacity of RUU (insts)
`-lsq:size <insts>`	- capacity of load/store queue (insts)
`-cache:dl1 <config>`	- level 1 data cache configuration
`-cache:dl1lat <cycles>`	- level 1 data cache hit latency

5. Hint

Sim-Outorder: Detailed Performance Simulator

`-cache:dl2 <config>`	- level 2 data cache configuration
`-cache:dl2lat <cycles>`	- level 2 data cache hit latency
`-cache:il1 <config>`	- level 1 instruction cache configuration
`-cache:il1lat <cycles>`	- level 1 instruction cache hit latency
`-cache:il2 <config>`	- level 2 instruction cache configuration
`-cache:il2lat <cycles>`	- level 2 instruction cache hit latency
`-cache:flush`	- flush all caches on system calls
`-cache:icompress`	- remap 64-bit inst addresses to 32-bit equiv.
`-mem:lat <1st> <next>`	- specify memory access latency (first, rest)
`-mem:width`	- specify width of memory bus (in bytes)
`-tlb:itlb <config>`	- instruction TLB configuration
`-tlb:dtlb <config>`	- data TLB configuration
`-tlb:lat <cycles>`	- latency (in cycles) to service a TLB miss

5. Hint

Sim-Outorder: Detailed Performance Simulator

`-res:ialu`	- specify number of integer ALUs
`-res:imult`	- specify number of integer multiplier/dividers
`-res:memports`	- specify number of first-level cache ports
`-res:fpalu`	- specify number of FP ALUs
`-res:fpmult`	- specify number of FP multiplier/dividers
`-pcstat <stat>`	- record statistic `<stat>` by text address
`-ptrace <file> <range>`	- generate pipetrace

5. Hint

Specifying the Branch Predictor

- specifying the branch predictor type

```
-bpred <type>
```

the supported predictor types are

`nottaken`	always predict not taken
`taken`	always predict taken
`perfect`	perfect predictor
`bimod`	bimodal predictor (BTB w/ 2 bit counters)
`2lev`	2-level adaptive predictor

- configuring bimodal predictors (when "`-bpred bimod`" is specified)

`-bpred:bimod <size>`	size of direct-mapped BTB


Thank you!