



Área Académica Ingeniera en Computadores

Arquitectura de Computadores I

Taller 3

Profesor:

Luis Chavarría Zamora

Estudiante:

Jose Ignacio Granados Marín

Grupo 1

IS-2022

Para la sección de la herramienta Wattch, luego de compilar el archivo test-fmath.c y comparar la ejecución fuera de orden con 1 unidad funcional por cada variable y con 8 unidades funcionales por cada variable (donde los datos que poseen el símbolo “-” representan las unidades funcionales con 1 unidad y los que poseen el símbolo “+” corresponden a las unidades funcionales con 8 unidades) se obtuvieron los siguientes resultados:

- Total Power Consumption:
 - -Total Power Consumption => 57.7939
 - +Total Power Consumption => 113.283
- Int ALU Power:
 - -Int ALU Power => 1.16503
 - +Int ALU Power => 9.32026
- FP ALU Power:
 - -FP ALU Power => 3.57026
 - +FP ALU Power => 28.5621
- avg_alu_power:
 - -avg_alu_power => 4.7353
 - +avg_alu_power => 37.8824
- avg_falu_power:
 - -avg_falu_power => 3.5703
 - +avg_falu_power => 28.5621
- total_power:
 - -total_power => 17518560.8055
 - + total_power => 20642221.5532

Cada uno de los datos anteriores concuerda con lo esperado debido a que, al aumentar la cantidad de unidades funcionales en simulador, el rendimiento del mismo se verá potenciado significativamente porque cuenta con más características, herramientas y métodos para llevar a cabo cada una de las funciones y operaciones solicitadas. Es por esta razón que los valores de Int ALU Power, FP ALU Power, avg_alu_power, avg_falu_power y total_power se ven beneficiados con esta modificación. Por el contrario, a mayor cantidad de dispositivos, mayor será el consumo de potencia debido a que se disponen de muchos más elementos que alimentar eléctricamente. Debido a lo anterior, el valor de Total Power Consumption con 8 unidades funcionales es aproximadamente el doble que el valor con 1 unidad funcional.

Por otra parte, al utilizar el simulador sim-profile con el mismo archivo test-fmath.c, se obtuvieron los siguientes resultados:

```

sim: ** simulation statistics **
sim_num_insn          93211 # total number of instructions executed
sim_num_refs          25075 # total number of loads and stores executed
sim_elapsed_time      1 # total simulation time in seconds
sim_inst_rate         93211.0000 # simulation speed (in insts/sec)

sim_inst_prof          # instruction profile
sim_inst_prof.array_size = 119
sim_inst_prof.bucket_size = 1
sim_inst_prof.count = 119
sim_inst_prof.total = 93210
sim_inst_prof.imin = 0
sim_inst_prof.imax = 119
sim_inst_prof.average = 783.2773
sim_inst_prof.std_dev = 2419.8355
sim_inst_prof.overflows = 0
# pdf == prob dist fn, cdf == cumulative dist fn
#          index      count      pdf
sim_inst_prof.start_dist
nop                15      0.02
j                  J        935      1.00
jal                J       1430      1.53
jr                 s       1489      1.60
jalr               d,s        36      0.04
beq                s,t,j    4304      4.62
bne                s,t,j    6762      7.25
blez               s,j        85      0.09
bgtz               s,j        67      0.07
bltz               s,j       435      0.47
bgez               s,j       559      0.60
bcif               j         44      0.05
bcit               j         18      0.02
lb                 t,o(b)     338      0.36
lbu                 t,o(b)   1119      1.20
lh                 t,o(b)      0      0.00
lhu                 t,o(b)     20      0.02
lw                 t,o(b)  12086     12.97
dlw                 t,o(b)     63      0.07
l.s                T,o(b)     328      0.35
l.d                T,o(b)     18      0.02
lwl                 t,o(b)      0      0.00
lwr                 t,o(b)      0      0.00
sb                  t,o(b)    830      0.89
sh                  t,o(b)      0      0.00
sw                  t,o(b)   9994     10.72
dsw                 t,o(b)     62      0.07
dsz                 o(b)       0      0.00
s.s                 T,o(b)    207      0.22
s.d                 T,o(b)     10      0.01
swl                 t,o(b)      0      0.00
swr                 t,o(b)      0      0.00
lb                  t,(b+d)     0      0.00
lbu                  t,(b+d)     0      0.00
lh                  t,(b+d)     0      0.00
lhu                  t,(b+d)     0      0.00
lw                  t,(b+d)     0      0.00
dlw                  t,(b+d)     0      0.00
l.s                  T,(b+d)     0      0.00
l.d                  T,(b+d)     0      0.00
sb                   t,(b+d)     0      0.00
sh                   t,(b+d)     0      0.00
sw                   t,(b+d)     0      0.00
dsw                  t,(b+d)     0      0.00
dsz                  (b+d)     0      0.00
s.s                  T,(b+d)     0      0.00
s.d                  T,(b+d)     0      0.00
l.s.r2               T,(b+d)     0      0.00
s.s.r2               T,(b+d)     0      0.00
lw.r2                t,(b+d)     0      0.00
sw.r2                t,(b+d)     0      0.00
add                  d,s,t       0      0.00
addi                 t,s,i       0      0.00
addu                  d,s,t    12544     13.46
addiu                 t,s,i    15779     16.93
sub                   d,s,t       0      0.00
subu                   d,s,t    2111      2.26
mult                  s,t        408      0.44
multu                 s,t    1259      1.35

```

Figura 1. Estadísticas de simulación del archivo test-fmath.c con el simulador sim-profile (parte 1).

```

div      s,t      0      0.00
divu     s,t      408     0.44
mfhi     d        1663    1.78
mthi     s        0      0.00
mflo     d        2075    2.23
mtlo     s        0      0.00
and      d,s,t    167     0.18
andi     t,s,u    2332    2.50
or       d,s,t    1249    1.34
ori      t,s,u    689     0.74
xor      d,s,t    68      0.07
xori     t,s,u    187     0.20
nor      d,s,t    242     0.26
sll      d,t,H    2316    2.48
sllv     d,t,s    1179    1.26
srl      d,t,H    523     0.56
srlv     d,t,s    1165    1.25
sra      d,t,H    86      0.09
srav     d,t,s    4       0.00
slt      d,s,t    367     0.39
slti     t,s,i    126     0.14
sltu     d,s,t    2993    3.21
sltui    t,s,i    552     0.59
add.s    D,S,T    1       0.00
add.d    D,S,T    64      0.07
sub.s    D,S,T    0       0.00
sub.d    D,S,T    32      0.03
mul.s    D,S,T    0       0.00
mul.d    D,S,T    51      0.05
div.s    D,S,T    0       0.00
div.d    D,S,T    4       0.00
abs.s    D,S      0       0.00
abs.d    D,S      0       0.00
mov.s    D,S      0       0.00
mov.d    D,S      92      0.10
neg.s    D,S      0       0.00
neg.d    D,S      1       0.00
cvt.s.d  D,S      0       0.00
cvt.s.w  D,S      0       0.00
cvt.d.s  D,S      4       0.00
cvt.d.w  D,S      9       0.01
cvt.w.s  D,S      2       0.00
cvt.w.d  D,S      12      0.01
c.eq.s   S,T      0       0.00
c.eq.d   S,T      35      0.04
c.lt.s   S,T      0       0.00
c.lt.d   S,T      19      0.02
c.le.s   S,T      0       0.00
c.le.d   S,T      8       0.01
sqrt.s   D,S      0       0.00
sqrt.d   D,S      0       0.00
syscall  18       0.02
break    B        0       0.00
lui      t,U      813     0.87
mfc1     t,S      144     0.15
dmfc1    t,S      2       0.00
cfc1     t,S      0       0.00
mtc1     t,S      153     0.16
dmtc1    t,S      0       0.00
ctc1     t,S      0       0.00
sim_inst_prof.end_dist

ld_text_base      0x00400000 # program text (code) segment base
ld_text_size      80384 # program text (code) size in bytes
ld_data_base      0x10000000 # program initialized data segment base
ld_data_size      12288 # program init'ed '.data' and uninit'ed '.bss' size in bytes
ld_stack_base     0x7fff0000 # program stack segment base (highest address in stack)
ld_stack_size     16384 # program initial stack size
ld_prog_entry     0x00400140 # program entry point (initial PC)
ld_environ_base   0x7fff8000 # program environment base address address
ld_target_big_endian 0 # target executable endian-ness, non-zero if big endian
mem.page_count    31 # total number of pages allocated
mem.page_mem      124k # total size of memory pages allocated
mem.ptab_misses   32 # total first level page table misses
mem.ptab_accesses 931627 # total page table accesses
mem.ptab_miss_rate 0.0000 # first level page table miss rate

```

Figura 2. Estadísticas de simulación del archivo test-fmath.c con el simulador sim-profile (parte 2).

El resultado anterior muestra un mapeo de la cantidad de instrucciones, con sus respectivos índices, contenidas dentro del archivo test-fmath.c, esto permite conocer y analizar más a profundidad el código en cuestión. Ahora bien, si se integrara el simulador de sim-profile con la herramienta de análisis de potencia Wattch, se podría realizar un cálculo de la cantidad unidades funcionales que dicho archivo necesita para operar con el menor consumo de potencia posible. Es decir, con base en el código que se desee ejecutar, se puede configurar y adaptar el simulador para que el CPU pueda realizar las tareas solicitadas con el mayor rendimiento posible.

Para la sección de simulación caché de este taller, cada una de las pruebas y resultados fueron ejecutados y extraídos del archivo test-printf.c.

Al utilizar el simulador sim-cache, se obtuvieron las siguientes estadísticas de simulación con la configuración por defecto de la memoria:

```
sim: ** simulation statistics **
sim_num_insn      6980362544 # total number of instructions executed
sim_num_refs      1610808883 # total number of loads and stores executed
sim_elapsed_time   365 # total simulation time in seconds
sim_inst_rate     19124280.9425 # simulation speed (in insts/sec)
l1l.accesses      6980362544 # total number of accesses
l1l.hits          6980350046 # total number of hits
l1l.misses        12498 # total number of misses
l1l.replacements  12242 # total number of replacements
l1l.writebacks    0 # total number of writebacks
l1l.invalidations 0 # total number of invalidations
l1l.miss_rate     0.0000 # miss rate (i.e., misses/ref)
l1l.repl_rate     0.0000 # replacement rate (i.e., repls/ref)
l1l.wb_rate       0.0000 # writeback rate (i.e., wrbks/ref)
l1l.inv_rate      0.0000 # invalidation rate (i.e., invs/ref)
d1l.accesses      1610810023 # total number of accesses
d1l.hits          1543699501 # total number of hits
d1l.misses        67110522 # total number of misses
d1l.replacements  67110266 # total number of replacements
d1l.writebacks    67110097 # total number of writebacks
d1l.invalidations 0 # total number of invalidations
d1l.miss_rate     0.0417 # miss rate (i.e., misses/ref)
d1l.repl_rate     0.0417 # replacement rate (i.e., repls/ref)
d1l.wb_rate       0.0417 # writeback rate (i.e., wrbks/ref)
d1l.inv_rate      0.0000 # invalidation rate (i.e., invs/ref)
ul2.accesses      134233117 # total number of accesses
ul2.hits          100677272 # total number of hits
ul2.misses        33555045 # total number of misses
ul2.replacements  33551749 # total number of replacements
ul2.writebacks    33551070 # total number of writebacks
ul2.invalidations 0 # total number of invalidations
ul2.miss_rate     0.2500 # miss rate (i.e., misses/ref)
ul2.repl_rate     0.2500 # replacement rate (i.e., repls/ref)
ul2.wb_rate       0.2499 # writeback rate (i.e., wrbks/ref)
ul2.inv_rate      0.0000 # invalidation rate (i.e., invs/ref)
itlb.accesses     6980362544 # total number of accesses
itlb.hits         6980362525 # total number of hits
itlb.misses       19 # total number of misses
itlb.replacements 0 # total number of replacements
itlb.writebacks   0 # total number of writebacks
itlb.invalidations 0 # total number of invalidations
itlb.miss_rate    0.0000 # miss rate (i.e., misses/ref)
itlb.repl_rate    0.0000 # replacement rate (i.e., repls/ref)
itlb.wb_rate      0.0000 # writeback rate (i.e., wrbks/ref)
itlb.inv_rate     0.0000 # invalidation rate (i.e., invs/ref)
dtlb.accesses     1610810023 # total number of accesses
dtlb.hits         1610285718 # total number of hits
dtlb.misses       524305 # total number of misses
dtlb.replacements 524177 # total number of replacements
dtlb.writebacks   524176 # total number of writebacks
dtlb.invalidations 0 # total number of invalidations
dtlb.miss_rate    0.0003 # miss rate (i.e., misses/ref)
dtlb.repl_rate    0.0003 # replacement rate (i.e., repls/ref)
dtlb.wb_rate      0.0003 # writeback rate (i.e., wrbks/ref)
dtlb.inv_rate     0.0000 # invalidation rate (i.e., invs/ref)
ld_text_base      0x00400000 # program text (code) segment base
ld_text_size      75424 # program text (code) size in bytes
ld_data_base      0x10000000 # program initialized data segment base
ld_data_size      13540 # program init'ed '.data' and uninit'ed '.bss' size in bytes
ld_stack_base     0x7ffffc000 # program stack segment base (highest address in stack)
ld_stack_size     16384 # program initial stack size
ld_prog_entry     0x00400140 # program entry point (initial PC)
ld_environ_base   0x7ffff8000 # program environment base address
ld_target_big_endian 0 # target executable endianness, non-zero if big endian
mem.page_count    524324 # total number of pages allocated
mem.page_mem      2097296k # total size of memory pages allocated
mem.ptab_misses   622616 # total first level page table misses
mem.ptab_accesses 31143545434 # total page table accesses
mem.ptab_miss_rate 0.0000 # first level page table miss rate
```

Figura 3. Configuración por defecto del simulador sim-cache.

Al analizar los resultados de la figura anterior, se puede extraer la siguiente información relacionada con el número de desaciertos de los caches L1 y L2:

- Desaciertos del cache L1:
 - `il1.misses => 12498`
 - `il1.miss_rate => 0.0000`
 - `dl1.misses => 67110522`
 - `dl1.miss_rate => 0.0417`
- Desaciertos del cache L2:
 - `ul2.misses => 33555845`
 - `ul2.miss_rate => 0.2500`

La figura y los datos anteriores corresponden a la información extraída sin ninguna modificación en los parámetros del simulador. Es decir, los resultados anteriores fueron obtenidos con la configuración por defecto del simulador `sim-cache`. Dicha configuración se muestra a continuación:

```
# -config                # load configuration from a file
# -dumpconfig           # dump configuration to a file
# -h                    false # print help message
# -v                    false # verbose operation
# -d                    false # enable debug message
# -i                    false # start in Dlite debugger
-seed                   1 # random number generator seed (0 for timer seed)
# -q                    false # initialize and terminate immediately
# -chkpt                <null> # restore EIO trace execution from <fname>
# -redir:sim            <null> # redirect simulator output to file (non-interactive only)
# -redir:prog           <null> # redirect simulated program output to file
-nice                   0 # simulator scheduling priority
-max:inst               0 # maximum number of inst's to execute
-cache:dl1              dl1:256:32:1:l # l1 data cache config, i.e., {<config>|none}
-cache:dl2              ul2:1024:64:4:l # l2 data cache config, i.e., {<config>|none}
-cache:il1              il1:256:32:1:l # l1 inst cache config, i.e., {<config>|dl1|dl2|none}
-cache:il2              dl2 # l2 instruction cache config, i.e., {<config>|dl2|none}
-tlb:itlb               itlb:16:4096:4:l # instruction TLB config, i.e., {<config>|none}
-tlb:dtlb               dtlb:32:4096:4:l # data TLB config, i.e., {<config>|none}
-flush                  false # flush caches on system calls
-cache:icompress        false # convert 64-bit inst addresses to 32-bit inst equivalents
# -pcstat               <null> # profile stat(s) against text addr's (mult uses ok)
```

Figura 4. Configuración por defecto del simulador `sim-cache`.

Ahora bien, si se modifican los parámetros de `cache:dl1`, `cache:dl2`, `cache:il1`, `cache:il2` y `flush` (donde los datos que poseen el símbolo “-” representan la configuración por defecto mientras que los que poseen el símbolo “+” corresponde a la nueva modificación) se obtienen los siguientes resultados:

➤ cache:dl1

Si se realiza la modificación de cache:dl1 dl1:512:64:2:1 y se compara con la configuración por defecto, se obtienen los siguientes datos del cache L1:

- -dl1.misses => 67110557
- +dl1.misses => 33555208
- -dl1.miss_rate => 0.0417
- +dl1.miss_rate => 0.0208

Al duplicar las características de la memoria de datos de nivel 1, se observa en los resultados anteriores que la cantidad de desaciertos y la tasa de desaciertos disminuye aproximadamente un 50%, lo cual implica una mejora en la memoria cache en cuestión. Dicho rendimiento incrementó debido a que en el archivo test-printf.c, en múltiples ocasiones, es necesario acceder a la memoria en búsqueda de datos necesarios para realizar ciertas operaciones, llevar el control de las iteraciones anidadas y acceder a los elementos de las diversas estructuras codificadas. A su vez, cualquier información operada debe ser almacenada nuevamente con el objetivo de actualizar los datos para futuros accesos a memoria. Por esta razón, al duplicar los procesos de lectura y escritura de datos, el rendimiento se beneficiará.

➤ cache:dl2

Si se realiza la modificación de cache:dl2 ul2:512:32:2:1 y se compara con la configuración por defecto, se obtienen los siguientes datos del cache L2:

- -ul2.misses => 33555823
- +ul2.misses => 67112164
- -ul2.miss_rate => 0.2500
- +ul2.miss_rate => 0.5000

Al disminuir las características de la memoria de datos de nivel 2 a la mitad, se observa en los resultados anteriores que la cantidad de desaciertos y la tasa de desaciertos incrementa aproximadamente un 50%, lo cual implica una disminución significativa en el rendimiento del CPU. Caso contrario a la modificación anterior, al encontrar una gran cantidad de lecturas y escrituras de datos, en múltiples ocasiones en el archivo test-printf.c, por parte de esta caché, estos procesos de acceso a memoria se verán afectados debido a que las líneas de caché no poseen las mismas dimensiones ni localidades espaciales que sus características por defecto.

➤ cache:il1

Si se realiza la modificación de cache:il1 il1:512:64:2:1 y se compara con la configuración por defecto, se obtienen los siguientes datos del cache L1:

- -il1.misses => 9210
- +il1.misses => 624

Al duplicar las características de la memoria de instrucciones de nivel 1, se observa en los resultados anteriores que la cantidad de desaciertos disminuyó aproximadamente 14.75 veces. Dicho rendimiento incrementó debido a que en el archivo test-printf.c, al haber codificadas tantas instrucciones en diversas funciones y en múltiples ciclos iterativos, la lectura de las mismas es un proceso que se ve afectado directamente por los bloques o líneas del caché. Y estos al ser incrementados aproximadamente un 6.775%, la cantidad de desaciertos es menor.

➤ cache:il2

Si se realiza la modificación de cache:il2 ul2:512:32:2:1 y se compara con la configuración por defecto, se obtienen los siguientes datos del cache L2:

- -ul2.misses => 33555823
- +ul2.misses => 33555208
- -ul2.miss_rate => 0.2500
- +ul2.miss_rate => 0.1656

Al disminuir las características de la memoria de instrucciones de nivel 2 a la mitad, se observa en los resultados anteriores que la cantidad de desaciertos disminuyó en 615 mientras que la tasa de desaciertos disminuyó un 0.0844, lo cual implica una pequeña mejora en el rendimiento del CPU. En este caso en particular, a pesar de haber limitado la memoria caché en cuestión, los resultados obtenidos potenciaron la ejecución del archivo test-printf.c y esto se debe a que dicho código presenta una pequeña variedad de instrucciones en cada uno de sus módulos. Por lo que, los procesos de lectura se ven beneficiados.

➤ flush

Si se realiza la modificación de flush true y se compara con la configuración por defecto, se obtienen los siguientes datos del cache L1 y L2:

- Desaciertos del cache L1:
 - -dl1.misses => 67110557
 - +dl1.misses => 67110592
- Desaciertos del cache L2:
 - -ul2.misses => 33555823
 - +ul2.misses => 33555908

La configuración de flush es la encargada de limpiar las secciones de las memorias caché de datos e instrucciones. Dicho proceso tiende a mejorar la velocidad del CPU. Inicialmente, la configuración por defecto de esta característica se encuentra en estado inactivo. Al habilitarla, por medio de la modificación en cuestión, se esperaría obtener una mejora significativa en el rendimiento. Sin embargo, ocurre todo lo contrario. Los datos anteriores evidencian un mínimo incremento en los desaciertos en ambas memorias caché y esto se debe, principalmente, a que en el archivo test-printf.c, a pesar de realizar un proceso de limpieza o flush de las memorias chache, la lectura y escritura, tanto de datos como de instrucciones, son procesos altamente recurrentes debido a la cantidad de ciclos iterativos. Por lo que, la limpieza de ambas memorias es innecesaria.