HPC Computer Architectures: Assembler Generation

AQTIVATE Training Workshop I

Dirk Pleiter

CST | EECS | KTH

December 2023

Assembler Generation

- Example GNU C-compiler: `gcc -S -O0 myprog.c`
  - Command will generate output file `myprog.s`
  - Usually better to start with zero-level optimization (`-O0`)
- Use cases:
  - Verification of sequence of instructions generated by compiler
  - Starting point for own assembler implementations

Assembler Generation Example

C-programm

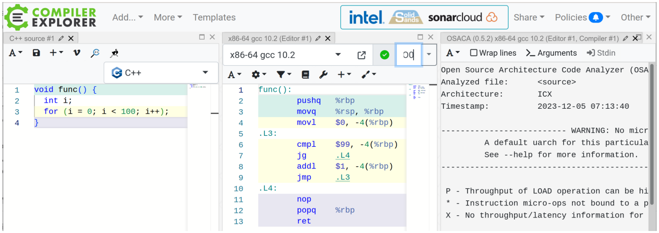
```
void func() {
  int i;
  for (i=0; i<100; i++);
}
```

Assembler:

```
         .file   "func.c"
         .text
.globl func
        .type   func, @function
func:
        pushl   %ebp
        movl    %esp, %ebp
        subl    $16, %esp
        movl    $0, -4(%ebp)
        jmp     .L2
.L3:
        addl    $1, -4(%ebp)
.L2:
        cmpl    $99, -4(%ebp)
        jle     .L3
        leave
        ret
```

Online Compiler Explorer

`https://godbolt.org/`



x86-64 gcc 10.2 O0

```
void func() {
  int i;
  for (i = 0; i < 100; i++);
}
```

```
func():
        pushq   %rbp
        movq    %rsp, %rbp
        movl    $0, -4(%rbp)
.L3:
        cmpl    $99, -4(%rbp)
        jg      .L4
        addl    $1, -4(%rbp)
        jmp     .L3
.L4:
        nop
        popq    %rbp
        ret
```

```
Open Source Architecture Code Analyzer (OSA
Analyzed file:      <source>
Architecture:       ICX
Timestamp:          2023-12-05 07:13:40

-------------------------- WARNING: No micr
         A default uarch for this particula
         See --help for more information.
-------------------------------------------

 P - Throughput of LOAD operation can be hi
 * - Instruction micro-ops not bound to a p
 X - No throughput/latency information for
```

Benefits:

- Interface makes reading assembler easier
- Easy to test different compilers and ISAs
- Integration of OSACA

Caveats:

- May need to explicitly define the target ISA variant (e.g. `-mavx512f`)