System-Level Programming

15 µC System Architecture – Preface

Peter Wägemann

Lehrstuhl für Informatik 4
Systemsoftware

Friedrich-Alexander-Universität
Erlangen-Nürnberg (FAU)

Summer Term 2025

http://sys.cs.fau.de/lehre/ss25

What does a µC understand? Compiler's Job?

- **µController cannot directly work on variables in memory**
- **µController can only make arithmetic operations on registers**
- assembly code contains **load-from** & **store-to sequences**
- compiler's job: decomposition of the program into smaller instructions that can be executed by a µController.
- example 1: decomposition of an expression

```
int a, b, c, d;

a = b + c * abs(d - 1);
```

```
int r0, r1, r2, r3;
int a, b, c, d;

r0 = b;
r1 = c;
r3 = d;
r3 -= 1;
r2 = abs(r3);
r1 *= r2;
r0 += r1;
a = r0;
```

`a, b, ...` : "variables in memory"

`r0, r1, ...` : "variables in registers"

Example: Decomposing Operations

- example 2: decomposition of a control structure

```
    if (n == 0) goto endif;
    i = 0;
    goto test;
loop:
    output();
    i++;
test:
    if (i != 10) goto loop;
endif:
```

```
    r0 = n;
    if (r0 == 0) goto endif;
    r0 = 0;
    i = r0;
    goto test;
loop:
    output();
    r0 = i;
    r0++;
    i = r0;
test:
    r0 = i;
    if (r0 != 10) goto loop;
endif:
```

Typical Operations on Registers

- abbreviated with `r1` or alternatively with `R1`
- here: `N` possible registers
- `rN = const;`
- `rN = var;`
- `rN op= const;`
- `rN op= rN;`
- `rN = func(...);`
- `var = rN;`
- `goto label;`
- `if (rN op const) goto label;`
- `if (rN op rM) goto label;`
- `return rN;`

Typical Assembly Instructions of µControllers

C Code	Mnemonic	
`rN++;`	`inc rN`	increment
`rN--;`	`dec rN`	decrement
`rN = const;`	`ldi rN, const`	load immediate
`rN = var;`	`ld rN, var`	load
`rN += const;`	`addi rN, const`	add immediate
`rN -= const;`	`subi rN, const`	subtract immediate
`rN += rM;`	`add rN, rM`	add
`rN -= rM;`	`sub rN, rM`	sub
`rN = func();`	`call func`	call function
`var = rN;`	`st var, rN`	store
`goto label;`	`jmp label`	jump
`if (rN == rM) goto label;`	`cmp rN, rM` `beq label`	compare branch if equal
...	...	

- all available instructions: see manual of processor/µController

C Code & Assembly Code

- example program:

C code		assembly code	
	uint8_t n;	10	
	uint8_t i;	11	
	...	...	
	r0 = n;	20	ld r0, 10
	if (r0 == 0) goto endif;	21	cmpi r0, 0
		22	beq 33
	r0 = 0;	23	ldi r0, 0
	i = r0;	24	st 11, r0
	goto test;	25	jmp 30
loop:	output();	26	call 70
	r0 = i;	27	ld r0, 11
	r0++;	28	inc r0
	i = r0;	29	st 11, r0
test:	r0 = i;	30	ld r0, 11
	if (r0 != 10) goto loop;	31	cmpi r0, 10
		32	bneq 26
endif:		33	
	...	...	...
	output(...)	70	...

Assembler's Job: Encoding Instructions

- example program:

assembly code		binary code
...		...
20	ld r0, 10	0a4f
21	cmpi r0, 0	a77f
22	beq 33	77bc
23	ldi r0, 0	87ee
24	st 11, r0	7439
25	jmp 30	30af
26	call 70	dd33
27	ld r0, 11	75ca
28	inc r0	9e88
29	st 11, r0	11f2
30	ld r0, 11	ad8f
31	cmpi r0, 10	54e1
32	bneq 26	98e4
...	...	...

- encoding of instructions is listed in µController's manual