



Universidade Tecnológica Federal do Paraná – UTFPR
Bacharelado em Ciência da Computação

BCC33B – Arquitetura e Organização de Computadores

Prof. Rogério A. Gonçalves
rogerioag@utfpr.edu.br

Aula 011

Decodificando

Aula de Hoje

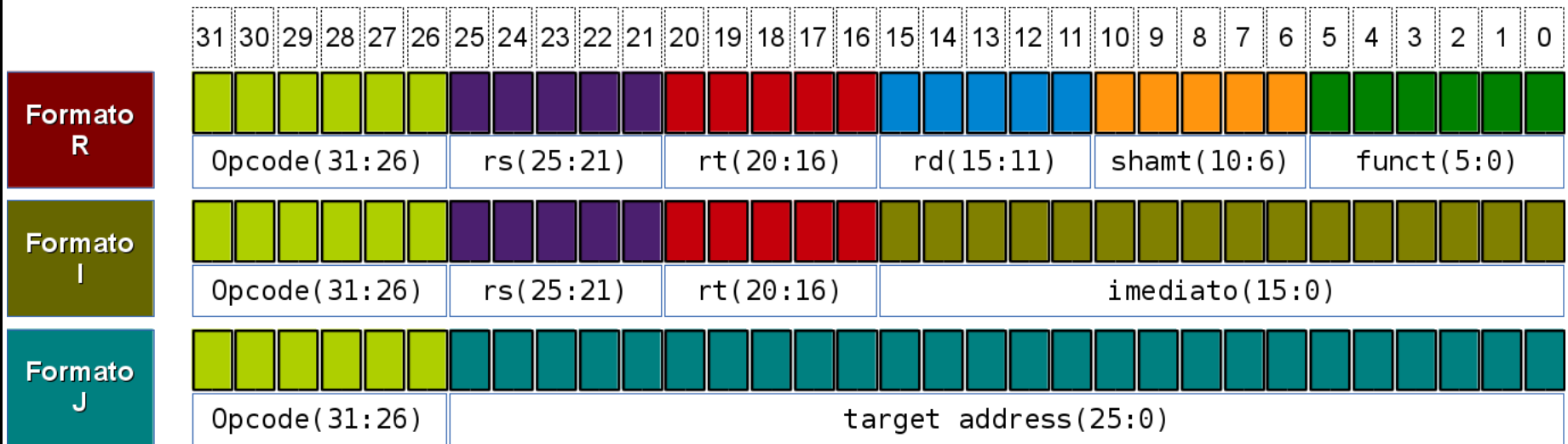
- Decodificação

Formato de Instruções

Name	Fields						Comments
Field size	6 bits	5 bits	5 bits	5 bits	5 bits	6 bits	All MIPS instructions 32 bits
R-format	op	rs	rt	rd	shamt	funct	Arithmetic instruction format
I-format	op	rs	rt	address/immediate			Transfer, branch, imm. format
J-format	op	target address					Jump instruction format

FIGURE 3.19 MIPS instruction formats in Chapter 3. Highlighted portions show instruction formats introduced in this section.

Campos do Formato de Instruções



Máscaras para o Formato R

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Formato R																																					
	Opcode (31:26)						rs (25:21)					rt (20:16)					rd (15:11)					shamt (10:6)					funct (5:0)										
Máscara opcode																																					
	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
	F						C					0					0					0					0										
Máscara rs	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
	0						3					E					0					0					0					0					
	0						0					1					F					0					0					0					
Máscara rt	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
	0						0					1					F					0					0					0					
	0						0					1					F					0					0					0					
Máscara rd	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0				
	0						0					0					F					8					0										
	0						0					0					F					8					0										
Máscara shamt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0					
	0						0					0					0					7					C										
	0						0					0					0					7					C										
Máscara funct	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1					
	0						0					0					0					0					3										
	0						0					0					0					0					3										

Lendo os campos

```
int mascaraOpCode = 0xFC000000;  
...  
int getOpCode(int ir) {  
    return (ir & mascaraOpCode) >> 26;  
}
```

Lendo os campos

```
int mascaraRs = 0x03E00000;
...
int getRs(int ir) {
    return (ir & mascaraRs) >> 21;
}
```

Lendo os campos

```
int mascaraRt =      0x001F0000;
...
int getRt(int ir) {
    return (ir & mascaraRt) >> 16;
}
```


Lendo os campos

```
int mascaraRd =      0x0000F800;
...
int getRd(int ir) {
    return (ir & mascaraRd) >> 11;
}
```

Lendo os campos

```
int mascaraShamt = 0x000007C0;
...
int getShamt(int ir) {
    return (ir & mascaraShamt) >> 6;
}
```

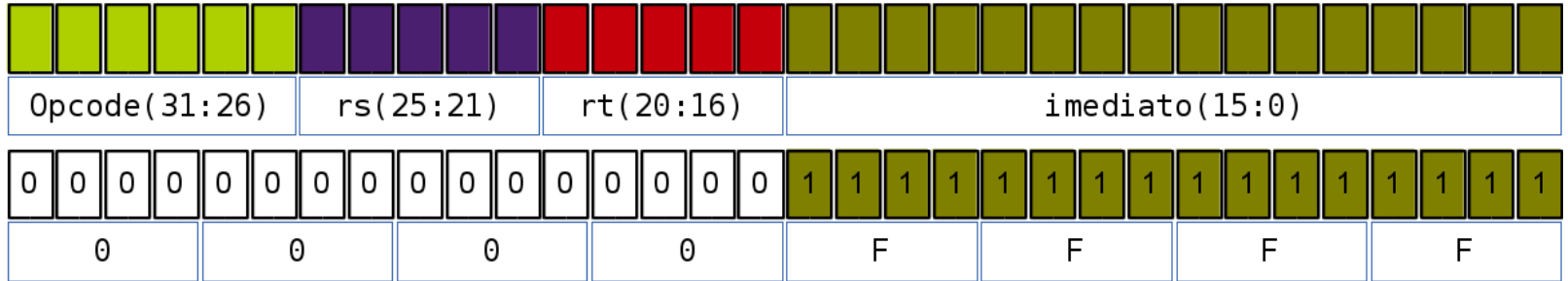
Lendo os campos

```
int mascaraFunct = 0x0000003F;  
...  
int getFunct(int ir) {  
    return (ir & mascaraFunct);  
}
```

Máscaras para o Formato I

Formato
I

Máscara
imediato

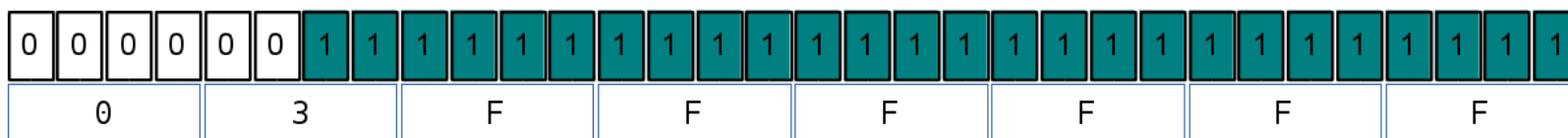


Máscaras para o Formato J

Formato
J



Máscara
address



Address = ir

Decoficador

```
// Decodifica uma instrução.
void decodificar(int ir) {
    switch (getOpCode(ir)) {
        case 0 : // 000000
            switch (getFunct(ir)) {
                case 32 : // 100000 -> add.
                    printf("add ");
                    printf("%s, ", registers[getRd(ir)]);
                    printf("%s, ", registers[getRs(ir)]);
                    printf("%s\n", registers[getRt(ir)]);
                    break;
                case 34 : // 100010 -> sub.
                    printf("sub ");
                    printf("%s, ", registers[getRd(ir)]);
                    printf("%s, ", registers[getRs(ir)]);
                    printf("%s\n", registers[getRt(ir)]);
                    break;
            }
        break;

        default : // outros casos.
            printf("Instrução não implementada. \n");
    }
}
```

Arquivo para leitura

```
GNU nano 2.2.6 Arquivo: exemplo.b
```

```
000000000000100010100000000100000
00000000000000000000000000000100000
0000000000000000000000000000000001
0000000000000000000000000000000010
00000010010100111000100000100010
```

```
^G Ajuda      ^O Gravar      ^R Ler o Arq ^Y Pág Anter ^K Recort Txt ^C Pos Atual
^X Sair      ^J Justificar ^W Onde está? ^V Próx Pág ^U Colar Txt ^T Para Spell
```

```
$ ./decodificador exemplo.b
```

```
Argumentos:
```

```
argc = 2
```

```
argv[0] = ./decodificador
```

```
argv[1] = exemplo.b
```

```
Linha: 00000000000100010100000000100000
```

```
Inteiro: 1130528
```

```
add $t0, $zero, $s1
```

```
Linha: 00000000000000000000000000000100000
```

```
Inteiro: 32
```

```
add $zero, $zero, $zero
```

```
Linha: 0000000000000000000000000000000001
```

```
Inteiro: 1
```

```
Linha: 0000000000000000000000000000000010
```

```
Inteiro: 2
```

```
Linha: 00000010010100111000100000100010
```

```
Inteiro: 39028770
```

```
sub $s1, $s2, $s3
```

Resumo da Aula de Hoje

Tópicos mais importantes:

- Decodificação

Entregar folha com:

- Nome
- RA
- Data de Hoje
- Resumo

Referências

- **Materiais de aula do Prof. João Angelo Martini do DIN-UEM.**
- **Livros: Patterson e Stalings.**