

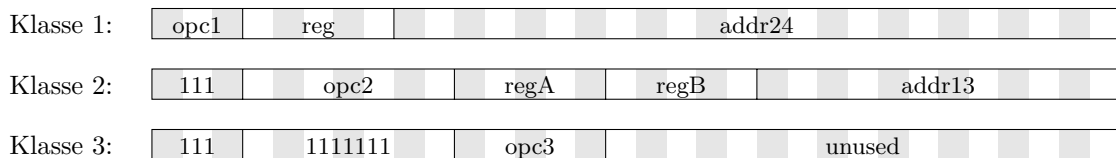
RS 11 (HA) zum 18.01.2013

Paul Bienkowski, Hans Ole Hatzel

18. Januar 2013

1. a) 30 b) 40 c) 50 d) 40 e) 40

2. Es gibt 3 Klassen von Befehlen, siehe Aufgabe, die jeweils voneinander unterscheidbar sein müssen.



Um alle benötigten 32 bit für Klasse 1 (3 bit Opcode + 5 bit Register + 24 bit Adresse) unterbringen zu können, muss für den Opcode ein fester Platz vorgesehen sein. Um die anderen Klassen unterscheiden zu können, werden hier die ersten 3 Bits auf 1 gesetzt, da dies der einzige fehlende Opcode in der ersten Klasse ist. Ähnliches gilt für die Unterscheidung der Klassen 2 und 3, jedoch könnte hier jeder andere freie Opcode aus Klasse 2 verwendet werden. Der Übersichtlichkeit halber wurde ebenfalls 1111111 gewählt.

Da für Klasse 2 zuerst 3 bit zur Unterscheidung von Klasse 1 nötig sind, 7 bit für den Opcode ($2^7 = 128 \leq 100$) und 2×5 bit für die Register verwendet werden, bleiben $32 \text{ bit} - (3 + 7 + 2 \cdot 5) \text{ bit} = 12 \text{ bit}$ für das Adressoffset.

3. a) 0000 1011 1001.
b) Nicht möglich, da der Abstand vom ersten 1-bit bis zum letzten 1-bit im Ergebnis 9 Stellen ist, und dies nicht im *imm8* dargestellt werden kann.
c) Nicht möglich, da eine Rotation um 29 Stellen erfordert wäre.
d) 1110 0110 0011.
e) 0010 0000 1001 *oder* 0011 0010 0100 *oder* 0100 1001 0000.

4. a) **0-Adress-Maschine:**

Befehl	Stack (TOS ist links)
PUSH E	E
PUSH D	D ; E
MUL	D * E
PUSH F	F ; D * E
ADD	F + D * E
PUSH C	C ; F + D * E
PUSH B	B ; C ; F + D * E
MUL	B * C ; F + D * E
PUSH A	A ; B * C ; F + D * E
SUB	A - B * C ; F + D * E
DIV	(A - B * C) / (F + D * E)
POP R	R = (A - B * C) / (F + D * E)

1-Adress-Maschine:

Befehl	Akkumulator/Bedeutung
LOAD D	Akku = D
MUL E	Akku = D * E
ADD F	Akku = D * E + F
STORE G	Akku = G = D * E + F
LOAD B	Akku = B
MUL C	Akku = B * C
STORE H	MEM[H] = B * C
LOAD A	Akku = A
SUB H	Akku = A - H = A - B * C
DIV G	Akku = (A - B * C) / G = (A - B * C) / (D * E + F)
STORE R	MEM[R] = (A - B * C) / (D * E + F)

2-Adress-Maschine:

Befehl	Bedeutung
MUL D, E	MEM[D] = MEM[D] * MEM[E] = D * E
ADD D, F	MEM[D] = MEM[D] + MEM[F] = D * E + F
MUL B, C	MEM[B] = MEM[B] * MEM[C]
SUB A, B	MEM[A] = MEM[A] - MEM[B] = A - B * C
DIV A, D	MEM[A] = MEM[A] / MEM[D] = (A - B * C) / (D * E + F)
MOV R, A	MEM[R] = MEM[A] = (A - B * C) / (D * E + F)

3-Adress-Maschine:

Befehl	Bedeutung
LOAD Y, D	Y = MEM[D]
LOAD Z, E	Z = MEM[E]
MUL Z, Y, Z	Z = Y * Z = D * E
LOAD Y, F	Y = MEM[F]
ADD Z, Z, Y	Z = Z + Y = D * E + F
LOAD X, D	X = MEM[B]
LOAD Y, E	Y = MEM[C]
MUL Y, X, Y	Y = X * Y = B * C
LOAD X, A	X = MEM[A]
SUB X, X, Y	X = X - Y = A - B * C
DIV X, X, Z	X = X / Z = (A - B * C) / (D * E + F)
STORE R, X	MEM[R] = X

b)	Maschine	Opcodes	Registernummern	Speicheradressen	Größe
	0-Adress	12 × 8 bit	0 × 4 bit	7 × 16 bit	208 bit
	1-Adress	11 × 8 bit	0 × 4 bit	11 × 16 bit	264 bit
	2-Adress	6 × 8 bit	0 × 4 bit	12 × 16 bit	240 bit
	3-Adress	12 × 8 bit	22 × 4 bit	7 × 16 bit	296 bit