

Fakultät: Informatik
 Fach: IT-Systeme / IT-Systeme 2
 Prüfer: Prof. Dr. F.J. Schmitt
 Prüfung: 12. Juli 2014

90 Minuten

Zugelassene Unterlagen: keine

Bearbeiten Sie die 6 Aufgaben ausschließlich auf den Aufgabenblättern (6 Seiten Angaben). Schreiben Sie Ihren vollständigen Namen auf jedes Blatt. Verwenden Sie zum Schreiben keinen Bleistift.

Sollte bei den Angaben eine Teilinformation fehlen oder fehlerhaft sein, so treffen Sie selbst eine geeignete Annahme und dokumentieren Sie sie.

Name:

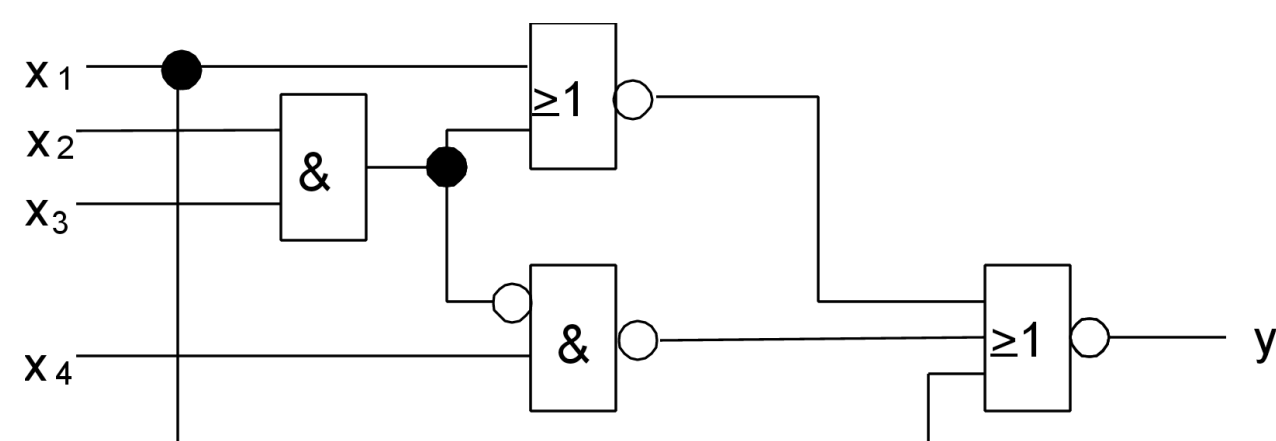
Aufgabe 1:

Matr.Nr:.....

a) Stellen Sie die KNF aus der folgenden Funktionstabelle auf:

x_1	x_2	x_3	Y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

b) Stellen Sie die Schaltungsgleichung der folgenden Schaltung auf:



Aufgabe 2:

- a) Skizzieren sie die CMOS-Realisierung eines NOR-Gatters mit zwei Eingängen x1 und x2 und dem Ausgang y.

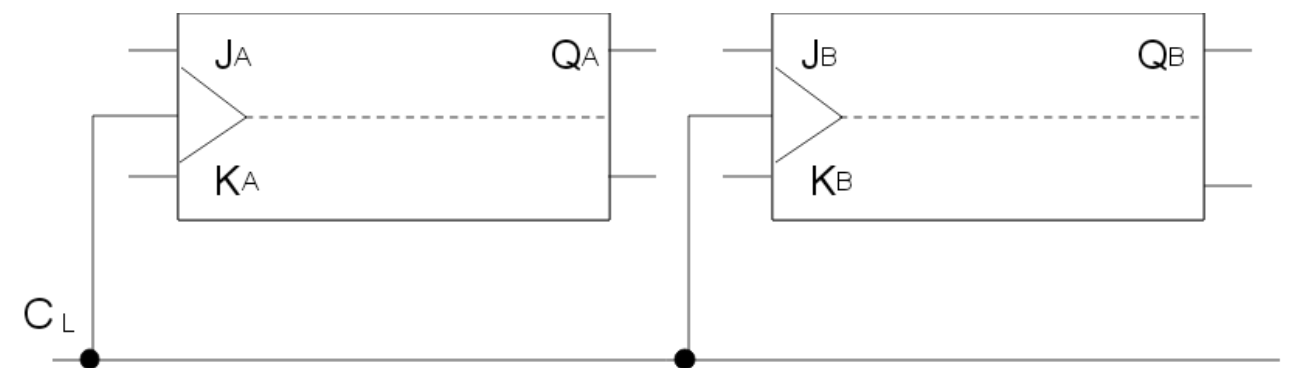
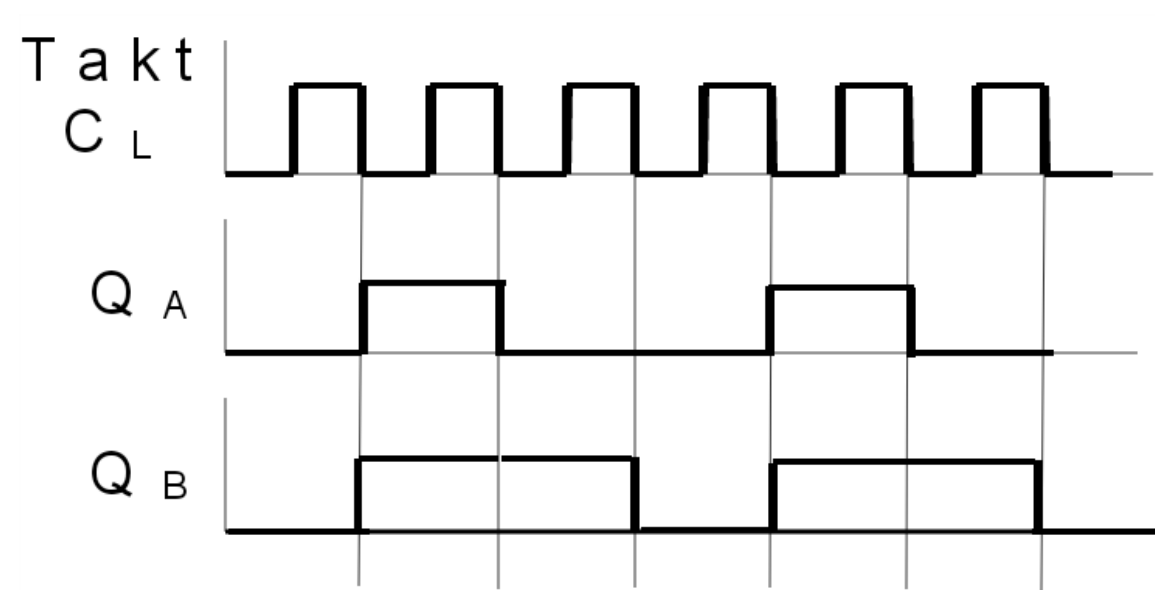
Hinweis: jeweils zwei p-Kanal und n-Kanal Transistoren, Versorgungsspannung (VDD) und Masse (Ground) werden verwendet;
beschriften sie die Transistoren entsprechend als p-, oder n-Kanal.

- b) Erläutern sie das Carry-Look-Ahead-Additionsverfahren am Beispiel der 8-Bit Addition von 43 und 57. Erläutern sie ihre Berechnung durch entsprechende Textanmerkungen.

43 =>	0	0	1	0	1	0	1	1
+ 57 =>	0	0	1	1	1	0	0	1

Aufgabe 3:

Aus zwei JK-Master-Slave-Flipflops soll eine synchrone Schaltung aufgebaut werden, die das folgende Impulsdiagramm liefert:



Leiten Sie aus dem Impulsdiagramm eine Wert-/Funktionstabelle ab, in der zu Q_A, Q_B, die notwendigen Eingangswerte J_A, K_A, J_B, K_B, enthalten sind. Berücksichtigen Sie dabei die Redundanzen.

Bestimmen Sie daraus mittels Karnaugh-Diagramm minimierte Gleichungen für J_A, K_A, J_B, K_B, und zeichnen Sie die sich ergebenden Verbindungen in das Schaltbild ein:

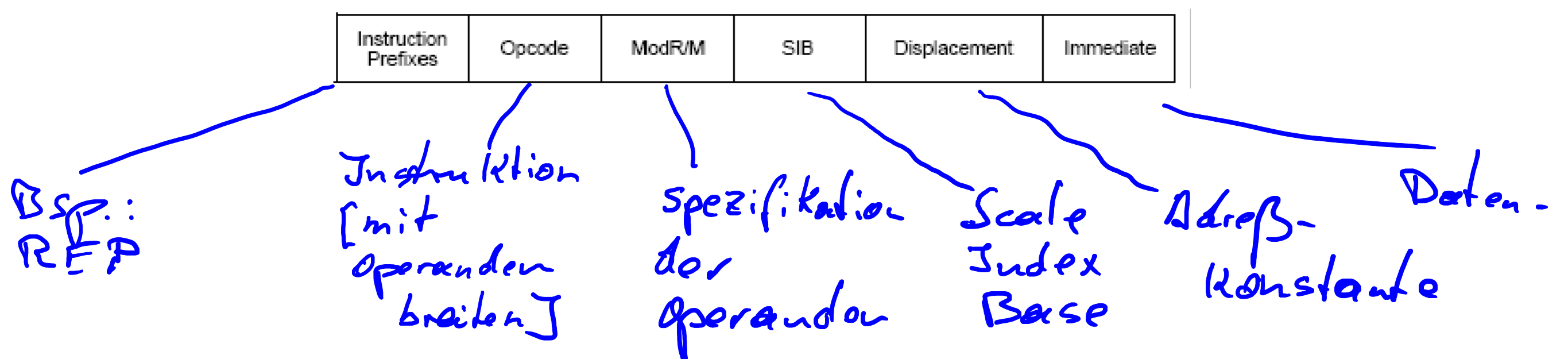
Aufgabe 4:

a) Geben Sie die Betragsdarstellung von 41 mit 8-Bit an.

b) Geben Sie die 2-er Komplementdarstellung von -41 in 8-Bit Darstellung an.

c) Geben Sie die Single-Precision Floating-Point Darstellung von -18,75 an.

d) Die nachfolgende Skizze gibt die grundsätzliche Instruktionscodierung für 80x86 an. Erläutern sie jeweils die Bedeutung der sechs Bestandteile.



Aufgabe 5:

Übersetzen Sie den Rumpf der nachfolgenden C-Funktion entsprechend der in der Vorlesung besprochenen Konventionen in 80x86 Inline-Assembler (Aufrufkonvention `_cdecl`).

```
int a[1000];
int i;

main() {
    for (i=0; i<1000; i++)
        a[i] = i;
}
```

Lösung:

```
main() {
    __asm{
```

`mov DWORD PTR [i], 0`

Bedingung:

`cmp DWORD PTR [i], 1000`

`JNL Ende`

`MOV EAX, DWORD PTR [i]`

`MOV DWORD PTR [i+EAX*4], EAX`

`JNC DWORD PTR [i]`

`JMP Bedingung`

Ende:

```
}
}
```

Aufgabe 6:

Übersetzen Sie den Rumpf der nachfolgenden C-Funktion entsprechend der in der Vorlesung besprochenen Konventionen in 80x86 Assembler (Aufrufkonvention `__declspec (naked)`).

```
int f (int a){
```

```
    int wert;
```

```
    wert = a;
```

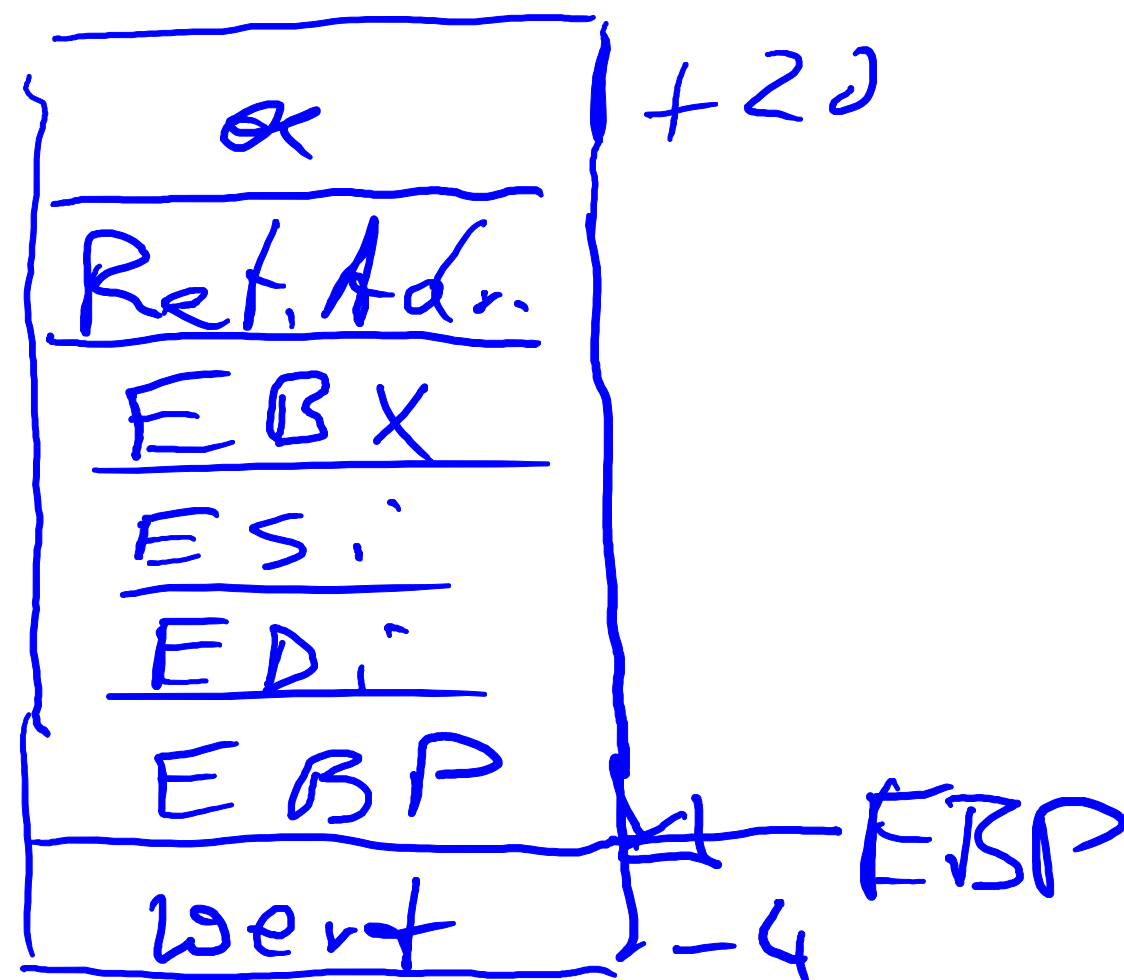
```
    if (a > 0)
```

```
        wert = 2*(a + f(a-1));
```

```
    return wert;
```

```
}
```

Skizze Stackframe angeben:



Lösung:

```
__declspec (naked) int f (int a) {
```

```
__asm{
```

```
    push EBX
```

```
    push ESI
```

```
    push EDI
```

```
    push EBP
```

```
    mov EBP, ESP
```

```
    sub ESP, 4
```

≡ Enter 4, 0

```
    mov EAX, DWORD PTR [EBP+20]
```

```
    mov DWORD PTR [EBP-4], EAX
```

```
    cmp DWORD PTR [EBP+20], 0
```

```
    jng Epilog
```

```
}
```

```
}
```

```

mov EAX, DWORD PTR [EBP+20]
dec EAX
push EAX
call f
add ESP, 4
add EAX, DWORD PTR [EBP+20]
imul EAX, 2
mov DWORD PTR [EBP-4], EAX

```

Epilog:

```

mov EAX, DWORD PTR [EBP-4]
add ESP, 4 ] = leave
pop EBP
pop EDI
pop ESI
pop EBX
ret

```

$$\text{wert} = 2 \times (a + f(a-1))$$