

Grundlagen Rechnertechnik

Prof. Dr. Peter Gerwinski

4. Dezember 2012

4 Der CPU-Stack

4.1 Implementation

- Speicherbereich als *Stack* reservieren
- Variable (typischerweise: Prozessorregister) als *Stack Pointer* reservieren
→ SP
- Assembler-Befehl `push foo`: $*SP++ = foo$;
- Assembler-Befehl `pop bar`: $bar = *--SP$;

Speziell: Unterprogramme

- `push IP`
`jmp foo` ← `mov #foo IP`
→ `call foo`
- `pop IP`
→ `ret`

4 Der CPU-Stack

4.2 Unterprogramme

Parameter:

- Prozessorregister
- CPU-Stack

Rückgabewert:

- Prozessorregister

Rückgabewert:

- Prozessorregister

5 Bit-Operationen

C-Operator	Verknüpfung
&	Und
	Oder
^	Exklusiv-Oder
~	Nicht

5 Bit-Operationen

C-Operator	Verknüpfung
&	Und
	Oder
^	Exklusiv-Oder
~	Nicht
<<	Verschiebung nach links
>>	Verschiebung nach rechts

5 Bit-Operationen

C-Operator	Verknüpfung	Anwendung
&	Und	Bits gezielt löschen
	Oder	Bits gezielt setzen
^	Exklusiv-Oder	Bits gezielt invertieren
~	Nicht	Alle Bits invertieren
<<	Verschiebung nach links	Maske generieren
>>	Verschiebung nach rechts	Bits isolieren

5 Bit-Operationen

C-Operator	Verknüpfung	Anwendung
&	Und	Bits gezielt löschen
	Oder	Bits gezielt setzen
^	Exklusiv-Oder	Bits gezielt invertieren
~	Nicht	Alle Bits invertieren
<<	Verschiebung nach links	Maske generieren
>>	Verschiebung nach rechts	Bits isolieren

Aufgabe: Schreiben Sie C-Funktionen, die ein „Array von Bits“ realisieren, z. B.

void set_bit (int i);	Bei Index <i>i</i> auf 1 setzen
void clear_bit (int i);	Bei Index <i>i</i> auf 0 setzen
int get_bit (int i);	Bei Index <i>i</i> lesen