



GO \$X,\$Y,\$Z

## Unterprogramme

Logisch zusammenhängende Befehlsfolge

- mehrfach verwendbar
- mit Namen aufrufbar

z.B.

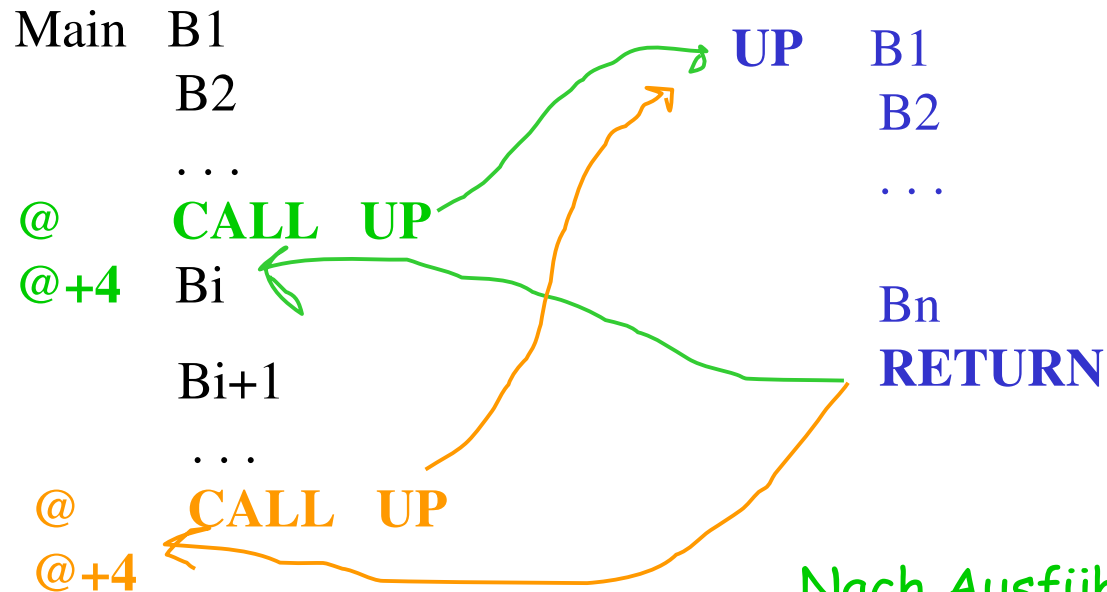
- Auswertung einer Formel
  - Ordnen von Elementen
- Suche des größten/kleinsten Elements
  - ...



GO \$X,\$Y,\$Z

## Unterprogramme

### Ablauf



Nach Ausführung  
Rückkehr an die Aufrufstelle



GO \$X,\$Y,\$Z

## Unterprogramme

### Beispiel

$$y = (x1 + x2) \uparrow 2$$

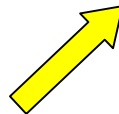
x1 in \$1, x2 in \$2, y nach \$3

Name des UP



```
Quadrat  ADD  $3,$1,$2  
          MUL  $3,$3,$3  
          RETURN
```

Rückkehr an die  
Aufrufstelle





GO \$X,\$Y,\$Z

## Unterprogramme

Realisierung von **CALL** mit GO

@ GO \$X,\$Y,0

\$X = @+4  
Returnadresse

Ziel= \$Y  
Name des UP



Nach dem Aufruf mit GO  
steht in \$X  
die Adresse  
des folgenden Befehls



Vor dem Aufruf  
Laden der Adresse des UP  
nach \$Y mit GETA



GO \$X,\$Y,\$Z

## Unterprogramme

Realisierung von **RETURN** mit GO

Während der Ausführung des Unterprogramms  
bleibt die Returnadresse (**@+4**) in **\$X** unverändert

**GO** dummy,**\$X,0**

springt in das aufrufende Programm  
an die beim CALL gespeicherte Adresse (**@+4**) zurück



GO \$X,\$Y,\$Z

## Unterprogramme

### Beispiel

Berechnung von

$$y = (x1 + x2) \uparrow 2 \quad \text{und} \quad z = (x3 + x4) \uparrow 2$$

Datensatz 1



X1

LOC

Data\_Segment

OCTA

127

X2

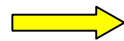
OCTA

539

Y

OCTA

Datensatz 2



X3

OCTA

257

X4

OCTA

153

Z

OCTA



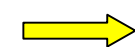
GO \$X,\$Y,\$Z

## Unterprogramme

### Beispiel

$$y = (x1 + x2)^{\uparrow 2}$$

1. CALL



```

LDO $1,X1
LDO $2,X2
GETA $0,Quadrat
GO $0,$0,0
Reta1 STO $3,Y

```

Quadrat

```

ADD $3,$1,$2
MUL $3,$3,$3
GO $0,$0,0

```



RETURN

$$z = (x3 + x4)^{\uparrow 2}$$

2. CALL



```

LDO $1,X3
LDO $2,X4
GETA $0,Quadrat
GO $0,$0,0
Reta2 STO $3,Z

```



GO \$X,\$Y,\$Z

## Unterprogramme

## Parameter-Übergabe

### Eingabeparameter

Werte mit denen das UP arbeitet

### Ausgabeparameter

Ergebnisse des UP

- ❖ Übergabe der **Werte** (call by value)
- ❖ Übergabe der **Adressen** (call by reference)



GO \$X,\$Y,\$Z

## Unterprogramme

### ❖ Wertübergabe in Registern

- vor CALL : Laden der Eingabeparameter

LDO \$1,X1

LDO \$2,X2

1. CALL

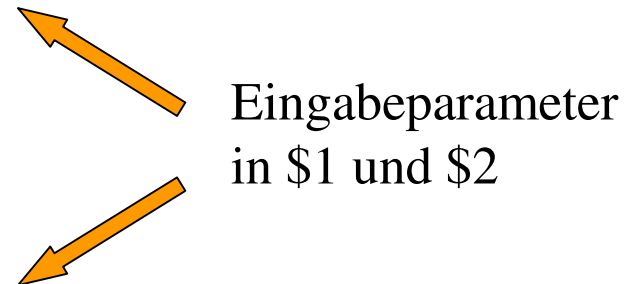
...

LDO \$1,X3

LDO \$2,X4

2. CALL

...





GO \$X,\$Y,\$Z

## Unterprogramme

### ❖ Wertübergabe in Registern

- im UP

**Quadrat**    ADD    \$3,\$1,\$2  
                  MUL    \$3,\$3,\$3  
                  GO     \$0,\$0,0

Eingabeparameter  
in \$1 und \$2



Ausgabeparameter in \$3





GO \$X,\$Y,\$Z

## Unterprogramme

### ❖ Wertübergabe in Registern

- nach RETURN : Speichern der Ausgabeparameter

1. CALL

STO **\$3**,Y

...

2. CALL

STO **\$3**,Z

Ausgabeparameter in \$3



GO \$X,\$Y,\$Z

## Unterprogramme

❖ Wertübergabe in Registern

😊 **Schneller Zugriff** auf die Werte im Unterprogramm

⚡ **Zahl der Parameter** muss bekannt sein  
(nicht für dynamische Strukturen  
z.B. arrays geeignet)

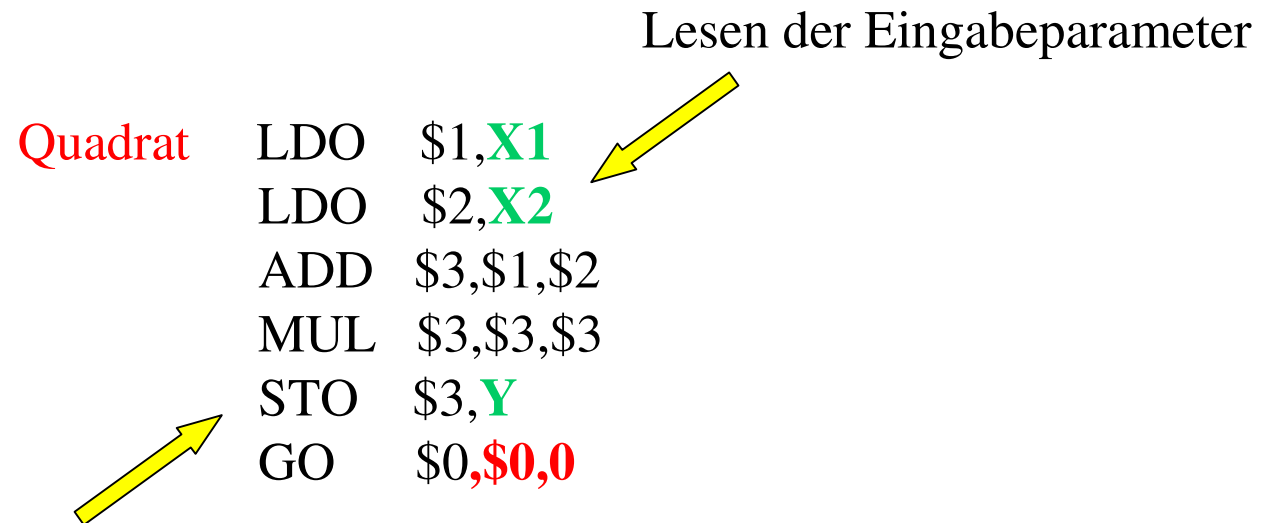
[MMIX - Programm](#)



GO \$X,\$Y,\$Z

## Unterprogramme

❖ Wertübergabe im Speicher



Speichern der Ausgabeparameter



GO \$X,\$Y,\$Z

## Unterprogramme

### ❖ Wertübergabe im Speicher

			LOC	Data_Segment
Datensatz 1	→	X1	OCTA	127
		X2	OCTA	539
		Y	OCTA	
Datensatz 2	→	X3	OCTA	257
		X4	OCTA	153
		Z	OCTA	

? {

⚡  
Problem: Umspeichern der Datensätze wegen fester Namen im UP bei RISC – Maschine nicht empfehlenswert!



GO \$X,\$Y,\$Z

## Unterprogramme

- ❖ Übergabe der **Anfangsadresse** der Parameter



			LOC	Data_Segment
Datensatz 1	→	<b>D1</b>	X1	OCTA 127
			X2	OCTA 539
			Y	OCTA
Datensatz 2	→	<b>D2</b>	X3	OCTA 257
			X4	OCTA 153
			Z	OCTA



GO \$X,\$Y,\$Z

## Unterprogramme

- ❖ Übergabe der Adressen im Register **point**

Vor dem 1. CALL:			LOC	Data_Segment
GETA	<b>point,D1</b>		<b>D1</b>	OCTA 127
				OCTA 539
				OCTA
Vor dem 2. CALL:				
GETA	<b>point,D2</b>		<b>D2</b>	OCTA 257
				OCTA 153
				OCTA



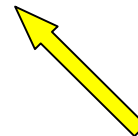
GO \$X,\$Y,\$Z

## Unterprogramme

### ❖ Übergabe der Anfangsadresse der Parameter

**Quadrat** LDO \$1,**point**,**0**  
LDO \$2,**point**,**8**  
ADD \$3,\$1,\$2  
MUL \$3,\$3,\$3  
STO \$3,**point**,**16**  
GO \$0,\$0,0

Anfangsadresse des  
aktuellen Parameterblocks  
im Register **point**



Zugriff auf Parameter  
mit Basisadresse und  
**Offset** als immediate Wert



GO \$X,\$Y,\$Z

## Unterprogramme

❖ Übergabe der Anfangsadresse der Parameter



Parameter müssen in einem **zusammenhängenden Speicherbereich** stehen



Übergabe von **dynamischen Strukturen** problemlos möglich

[MMIX - Programm](#)



GO \$X,\$Y,\$Z

## Unterprogramme mit GO

- Das Register mit der Returnadresse darf im Unterprogramm **nicht überschrieben** werden!!!
- Geschachtelte Aufrufe von UP müssen **Verschiedene Register** für die Returnadresse verwenden.
- Rekursive Aufrufe brauchen einen **Stack** (Last-In-First-Out)



GO \$X,\$Y,\$Z

## Unterprogramme

Up-Aufruf mit automatischem Stack : **PUSHJ, PUSHGO**

