Adressierung und Befehlsfolgen (4)

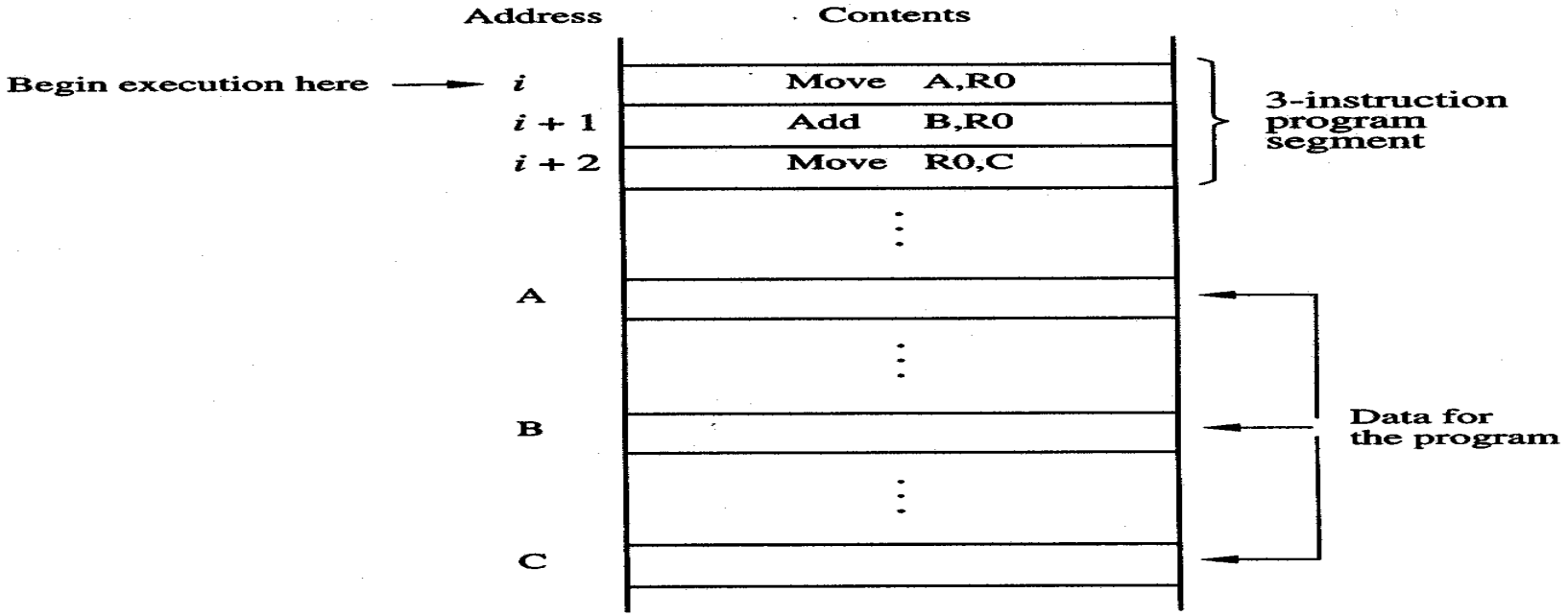
1.2 Befehlsausführung

2 Phasen:

- Befehlsholphase (instruction fetch)
- Befehlausführungsphase (instruction execute)

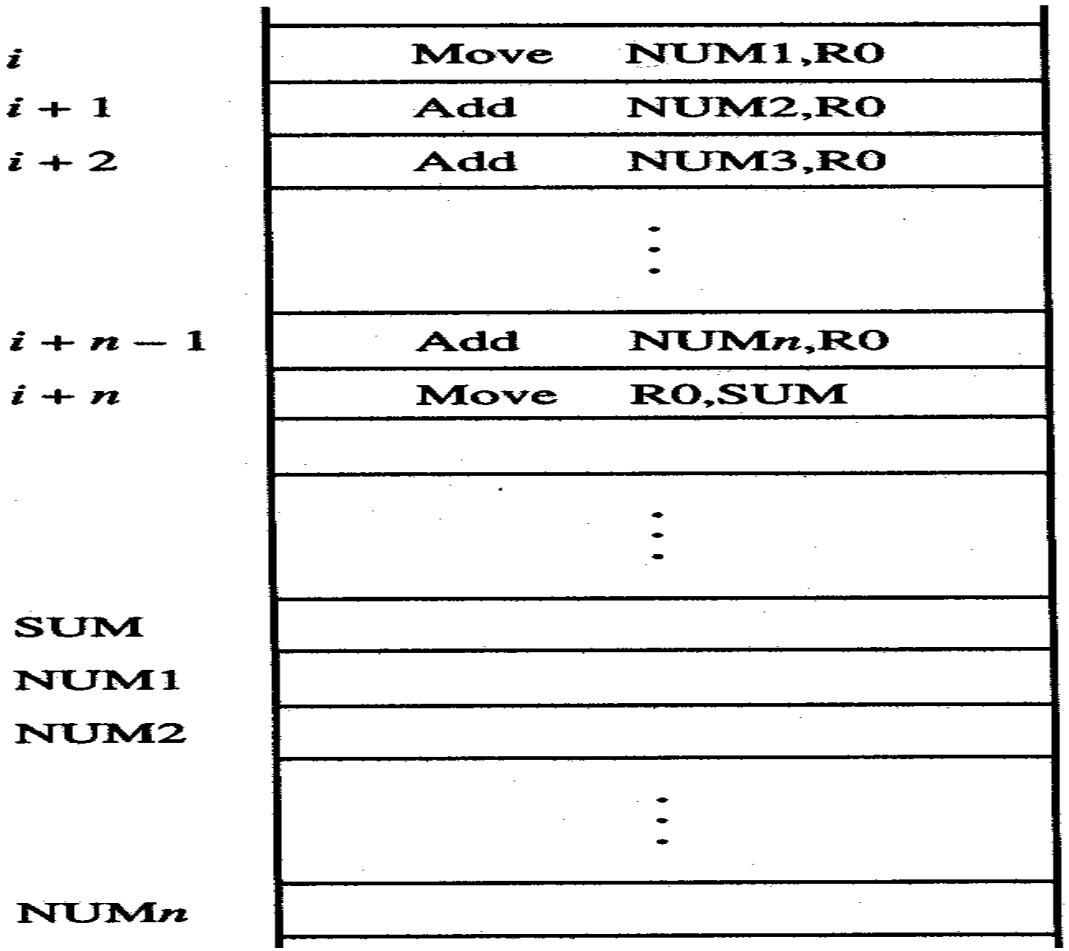
1.2.1 lineare Befehlsfolge (straight-line sequencing)

Ein im Speicher geladenes Programm für C ← [A] + [B]



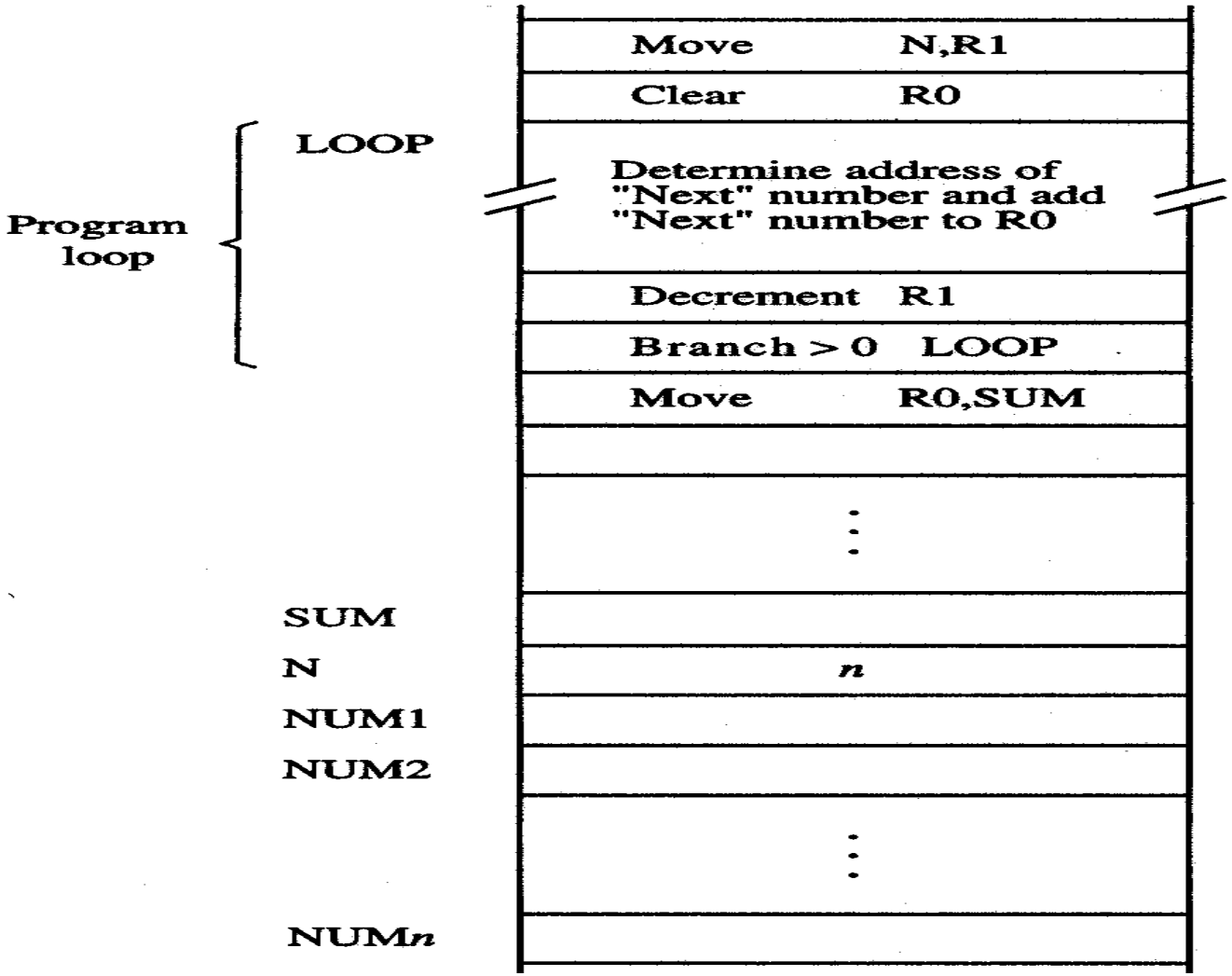
Adressierung und Befehlsfolgen (5)

Ein Programm zur Addition von n Zahlen



Adressierung und Befehlsfolgen (6)

Addition von n Zahlen mit Hilfe einer Schleife und einer Befehlsverzweigung (branching)



Adressierung und Befehlsfolgen (7)

1.2.4 Condition Codes

Vier häufig benutzte bits (flags) in einem CC (Status) - Register

N (negativ)	gesetzt auf 1, wenn Resultat negativ, sonst 0
Z (null)	gesetzt auf 1, wenn Resultat null, sonst 0
V(overflow)	gesetzt auf 1, wenn arithm. overflow auftritt, sonst 0
C (Übertrag)	gesetzt auf 1, wenn Übertag auftritt, sonst 0

1.3 Adressmodi

Der Begriff „Adressmodus“ bezieht sich auf die Art und Weise, in welcher der Operand eines Befehls spezifiziert ist.

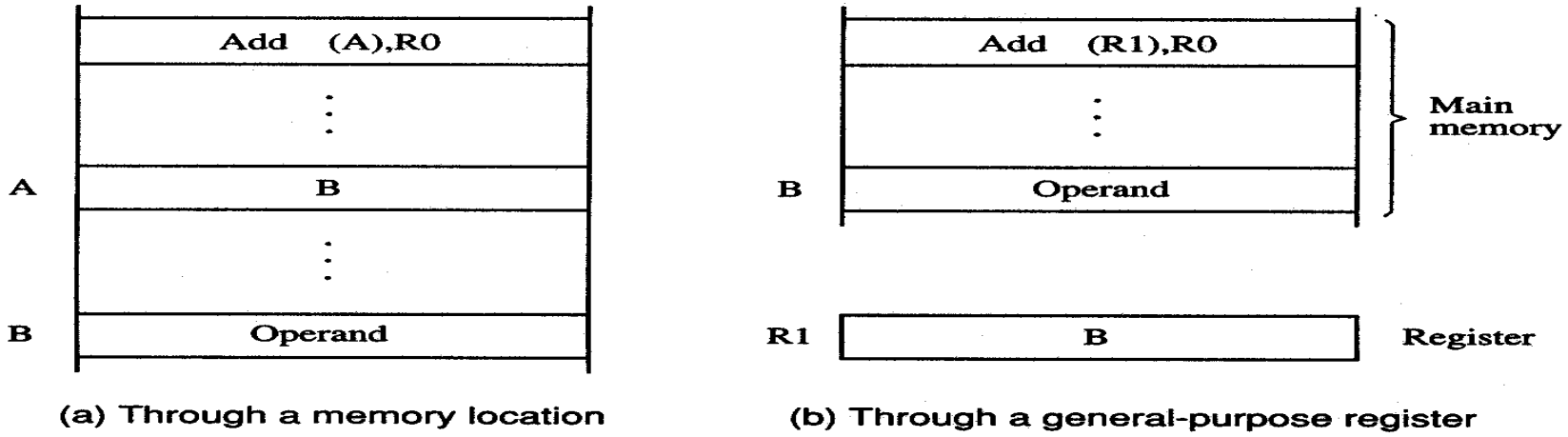
Bisher angewandt wurden

•*absolute mode (direkte Adressierung)*
Die effektive Speicheradresse der Operanden steht im Operandenteil des entsprechenden Befehls

•*register mode (registerdirekte Adressierung)*
Die effektive Registeradresse der Operanden steht im Operandenteil des entsprechenden Befehls

Adressierung und Befehlsfolgen (9)

Prinzip der indirekten Adressierung

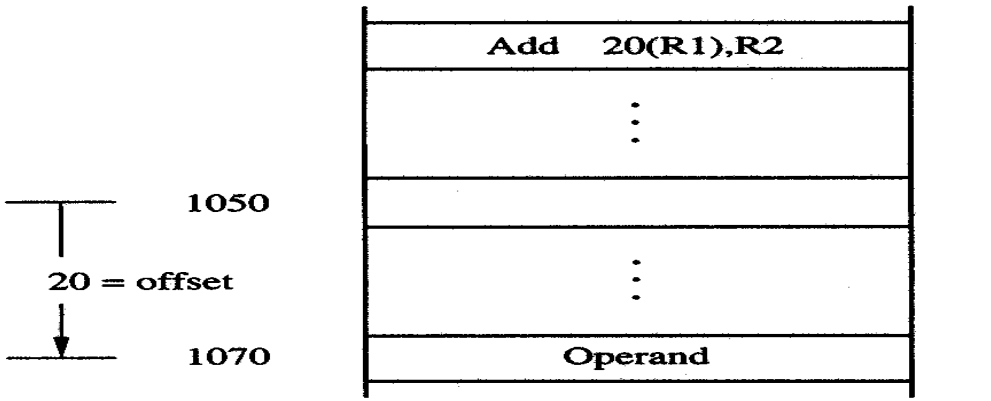


Awendung der indirekten Adressierung auf die Addition von n Zahlen

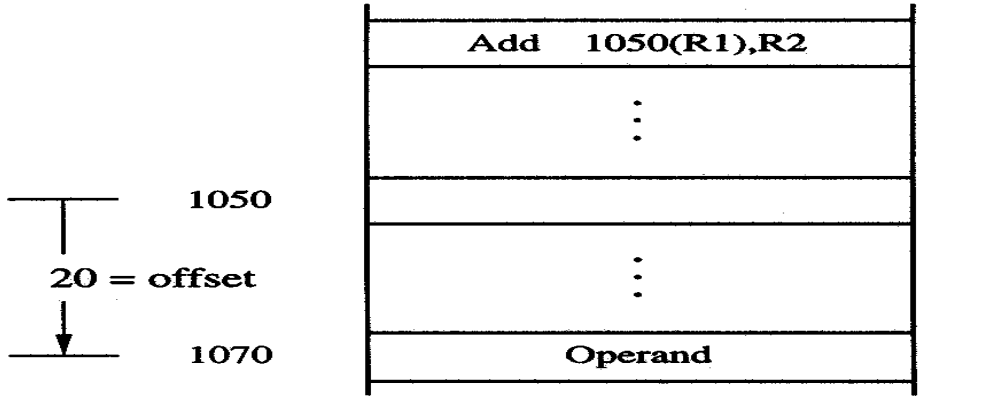
Address	Contents		
	Move	N,R1	Initialization
	Move	#NUM1,R2	
	Clear	R0	
LOOP	Add	(R2),R0	
	Increment	R2	
	Decrement	R1	
	Branch > 0	LOOP	
	Move	R0,SUM	

Adressierung und Befehlsfolgen (10)

Zwei Möglichkeiten zur Realisierung des Indexmodus



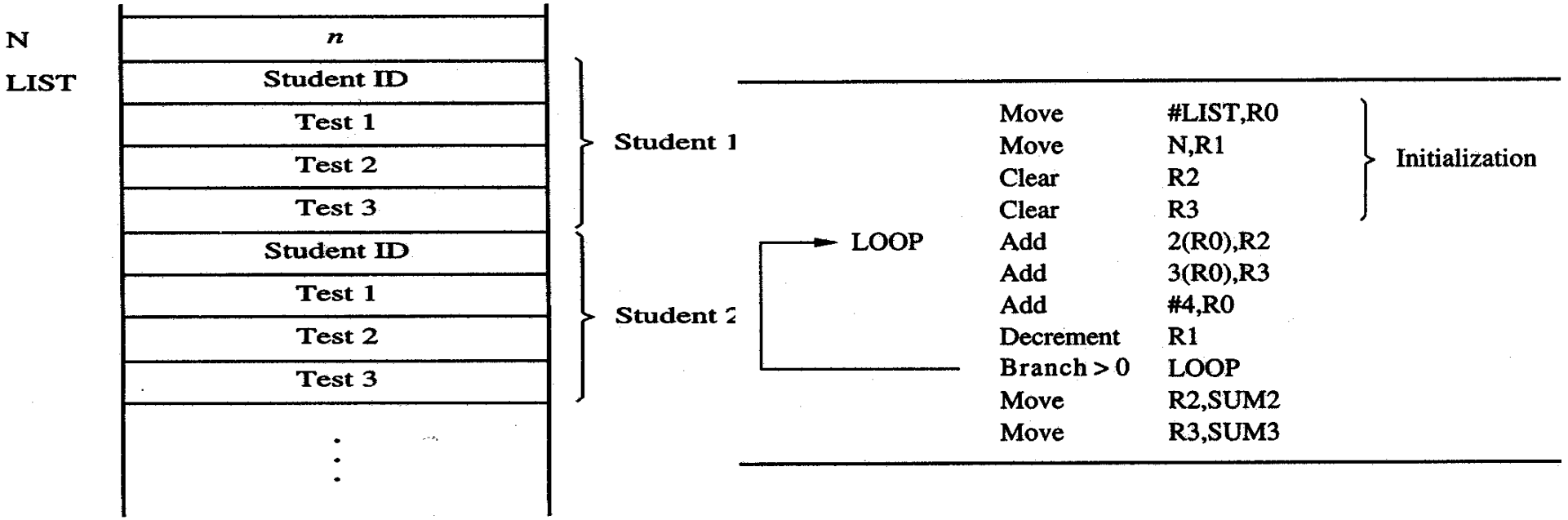
(a) Offset is given as a constant



(b) Offset is in the index register

Adressierung und Befehlsfolgen (11)

Einfaches Beispiel für die Anwendung der Indexadressierung



Anwendung des autoincrem. mode auf das Beispiel der indir. Adressierung