## Tutorium 02.12.2021

Mittwoch, 1. Dezember 2021

## Kodierung der Instruktionswörter

Jeder ARM-Assemblerbefehl wird nach diesem Schema in genau ein 32-Bit-Instruktionswort kodiert:

3 3 2 2 1 0 9 8	2 7	2 6		2						1 1 8 7	1 6	1 5	1 4	1 3	1 2	1	1	0	0 8	0	0	0	0	0 0 0 0	0	Befehlstyp
Bedingung	0	0	1	C	Opcode			5	Rn			Rd				2. Operand									i	Data Processing
Bedingung	0	0	0	0	0	0	Α	S		Rd			Rn				R	s		1	0	0	1	Rm		Multiply
Bedingung	0	1	1	P	U	В	w	L		Rn			R	d					Offset				t			Single Data Transfer
Bedingung	1	0	0	P	U	В	W	L		Rn						Registerliste										Block Data Transfer
Bedingung	0	0	0	P	U	1	w	L		Rn			R	d		0	ffs	et	1	1	5	Н	1	Offset 2	2	Halfword Trans Imm
Bedingung	0	0	0	P	U	0	W	L		Rn			R	d		0	0	0	0	1	S	Н	1	Rm		Halfword Trans Reg
Bedingung	1	0	1	L		relative Zieladresse																Branch				
Bedingung	0	0	0	1	0	0	1	0	1	1 1	1	1	1	1	1	1	0	0	0	1	S	Н	1	Rn		Branch Exchange
Bedingung	1	1	1	1	SWI-Nummer (vom Prozessor ignoriert)														Software Interrupt							

Einige ARM-Prozessoren unterstützen zusätzlich eine kompaktere Kodierung, die 16- und 32-Bit-Worte mischt. Dieser Thumb- und Thumb-2-Kode ist meist kürzer, aber langsamer (und nicht Stoff dieser Lehrveranstaltung).

12 -bils for dischles laded

8-6:1: Zahl

4-6:15: Barrel-shiften (ROR) 2 = 16 Moslichheiten {0,2,7,6,8,10,17,74 16,11,10,22, 24, 26,28, 30}

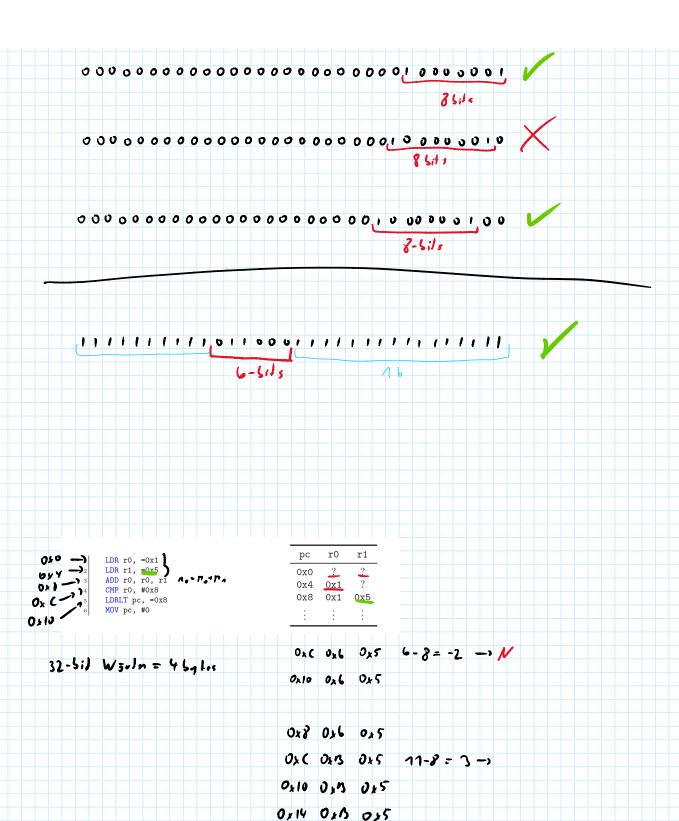
5: 1: 31 30 7 27 27 26 25 24 23 12 27 20 47 18 12 16 15 14 13 11 11 10 7 9 7 6 5 6 2 2 1 0 

(864) 10 = (11011,00000) 2

ROR  $(x_{31}, \ldots, x_0)$ 

 $x_0$ 

4 non (32-4) = 28



OND DIE DYS