



Verfasser:  
D. Gachet / HTA-FR - Telekommunikation

HTA-FR – Kurs Telekommunikation

**Embedded systems 1 und 2**  
Interne Architektur (Abstrakt)

Klasse T-2 // 2017-2018



# Interne Register



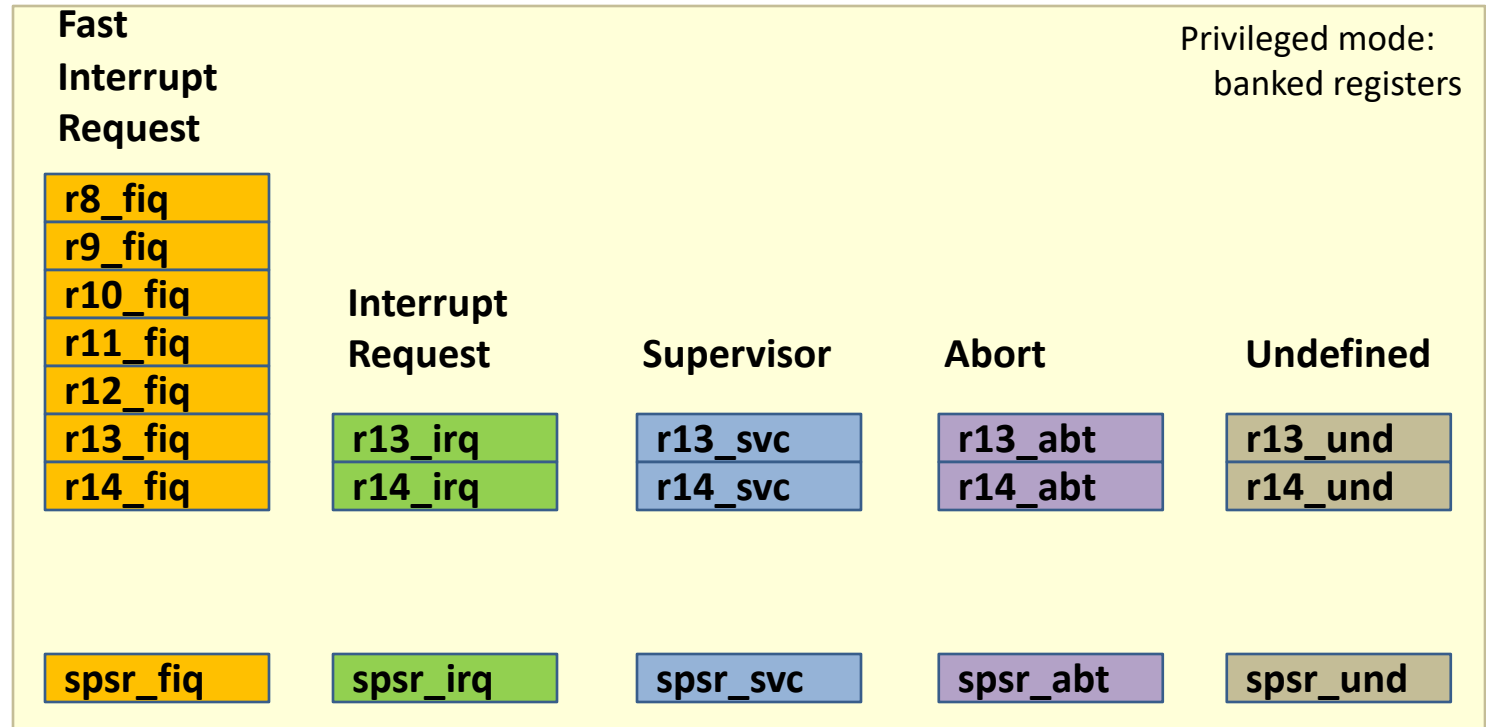
## User & System

r0
r1
r2
r3
r4
r5
r6
r7
r8
r9
r10
r11
r12
r13(sp)
r14(lr)
r15(pc)

cpsr

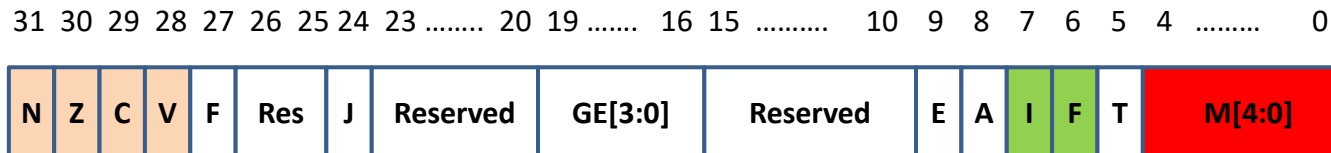
37 interne Register mit 32 Bit, davon:

- ❖ 15 allgemeine Register (R0 – R14)
- ❖ 1 Programmzähler (PC – *program counter*)
- ❖ 1 Statusregister für den laufenden Status (CPSR)
- ❖ 15 spezifische Register für die verschiedenen Modi (*banked registers*)
- ❖ 5 Statusregister für die Sicherung (SPSR)





Auf das Statusregister kann aus allen Funktionsmodi des Prozessors zugegriffen werden. Es enthält die Bedingungsmarken (*flags*), die Steuerbit für Unterbrechungen (Interrupts), die Steuerbit für den Prozessor-Funktionsmodus und weitere Bit, die sich auf die Prozessorfunktionen beziehen.



## The "*condition code flags*"

The N, Z, C, and V (Negative, Zero, Carry and oVerflow) bits are collectively known as the *condition code flags*, often referred to as *flags*. The *condition code flags* in the CPSR can be tested by most instructions to determine whether the instruction is to be executed.

## The "*interrupt disable bits*"

I, and F are the interrupt disable bits:

I bit Disables IRQ (Interrupt Request) interrupts when it is set.

F bit Disables FIQ (Fast Interrupt) interrupts when it is set.

## The "*mode*" bits

M[4:0] are the mode bits. These determine the mode in which the processor operates (usr, sys, fiq, irq, svc, abt, und)



- ▶ **N : Negative:** wenn  $N==1$ , negative Zahl.  
Dies ist das Vorzeichenbit der Zahlen im 2er-Komplement.
- ▶ **Z : Zero:** wenn  $Z==1$ , Zahl == 0.  
Im Allgemeinen mit Zählern benutzt, um Schleifen auszuführen.
- ▶ **C : Carry:** wenn  $C==1$ , report.  
Im Allgemeinen mit Zählern benutzt, um Schleifen auszuführen (Zahlen ohne Vorzeichen).
- ▶ **V : oVerflow:** wenn  $V==1$ , Kapazitätsüberschreitung (Überlauf).  
Hauptsächlich bei arithmetischen Operationen im 2er-Komplement (Zahlen mit Vorzeichen) verwendet.



# Codierung des Befehlssatzes\*)



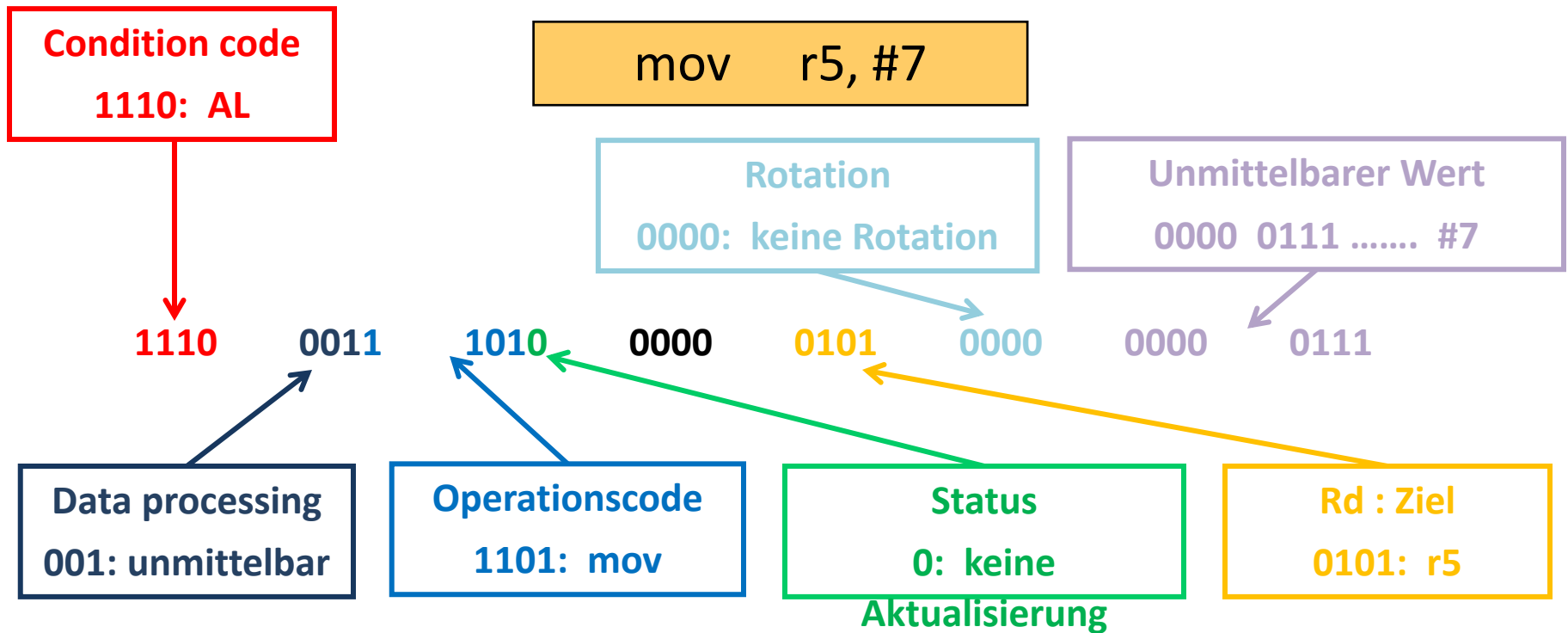
\*) siehe Kapitel A3.1 von 01\_ARM\_Architecture\_Reference\_Manual.pdf

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																							
Data processing immediate shift	cond [1]	0	0	0	opcode				S	Rn				Rd				shift amount				shift		0	Rm														
Miscellaneous instructions: See Figure A3-4	cond [1]	0	0	0	1	0	x	x	0	x x x x x x x x x x x x x x x x x x																						0	x x x x						
Data processing register shift [2]	cond [1]	0	0	0	opcode				S	Rn				Rd				Rs				0	shift		1	Rm													
Miscellaneous instructions: See Figure A3-4	cond [1]	0	0	0	1	0	x	x	0	x x x x x x x x x x x x x x x x x x																						0	x x		1	x x x x			
Multiplies: See Figure A3-3 Extra load/stores: See Figure A3-5	cond [1]	0	0	0	x x x x x x				x x x x x x x x x x x x x x x x x x																						1	x x		1	x x x x				
Data processing immediate [2]	cond [1]	0	0	1	opcode				S	Rn				Rd				rotate				immediate																	
Undefined instruction	cond [1]	0	0	1	1	0	x	0	0	x x x x x x x x x x x x x x x x x x																													
Move immediate to status register	cond [1]	0	0	1	1	0	R	1	0	Mask				SBO				rotate				immediate																	
Load/store immediate offset	cond [1]	0	1	0	P	U	B	W	L	Rn				Rd				immediate																					
Load/store register offset	cond [1]	0	1	1	P	U	B	W	L	Rn				Rd				shift amount				shift		0	Rm														
Media instructions [4]: See Figure A3-2	cond [1]	0	1	1	x x x x x x				x x x x x x x x x x x x x x x x x x																						1	x x x x							
Architecturally undefined	cond [1]	0	1	1	1	1	1	1	1	x x x x x x x x x x x x x x				1 1 1 1				x x x x																					
Load/store multiple	cond [1]	1	0	0	P	U	S	W	L	Rn				register list																									
Branch and branch with link	cond [1]	1	0	1	L	24-bit offset																																	
Coprocessor load/store and double register transfers	cond [3]	1	1	0	P	U	N	W	L	Rn				CRd				cp_num				8-bit offset																	
Coprocessor data processing	cond [3]	1	1	1	0	opcode1				CRn				CRd				cp_num				opcode2		0	CRm														
Coprocessor register transfers	cond [3]	1	1	1	0	opcode1				L	CRn				Rd				cp_num				opcode2		1	CRm													
Software interrupt	cond [1]	1	1	1	1	swi number																																	
Unconditional instructions: See Figure A3-6	1	1	1	1	x x																																		

Figure A3-1 ARM instruction set summary



# Beispiel eines einfachen Befehls



Maschinencode

0xE3A05007

Aktion

Speichern des Wertes 7 im Register R5