

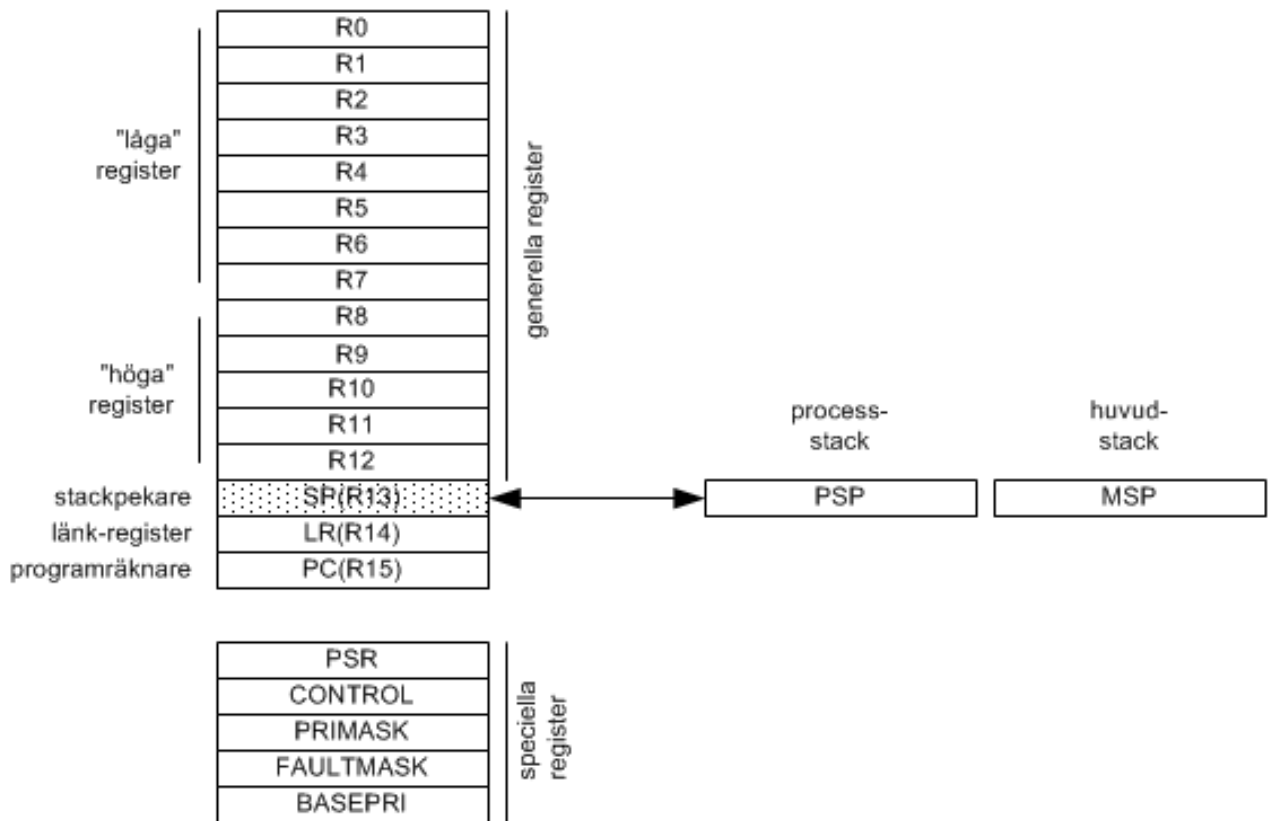
Quick Guide

Laborationsdator MD407 med tillbehör

Detta häfte får användas under tentamen i kurserna
"Programmering av inbyggda system"
"Maskinorienterad programmering"
under förutsättning att inga anteckningar gjorts häri.

Institutionen för Data och informationsteknik
Chalmers 2016

REGISTERUPPSÄTTNING



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	APSR
N	Z	C	V	Q													ISR_NUMBER										IPSR					
										ICI	IT											ICI						EPSR				

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	CONTROL
																										F	S	P				

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	PRIMASK
																										M						

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	FAULTMASK
																										M						

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	BASEPRI
														PRIORITY																		

R0-R12: Generella 32-bitars register.

R13: Stackpekare, i själva verket två olika register där inställningen i CONTROL-registret bestämmer vilket av registren PSP (process stack pointer) eller MSP (master stack pointer)

R14: Länkregister, i detta register sparas återhopsadressen vid BL (Branch and Link)instruktionen.

R15: Programräknaren

PSR: Program Status Register

Är i själva verket tre olika register

APSR (Application Program Status Register) innehåller statusbitar från operationer.

IPSR (Interrupt Program Status Register)

ISR_NUMBER: Är antingen 0, dvs. inget avbrott, eller indikerar det aktiva avbrottet.

EPSR (Execution Program Status Register)

Registret innehåller biten Thumb state och exekveringstillståndet för antingen:

- *If-Then (IT)* instruktionen
- *Interruptible-Continuable Instruction (ICI)* fält för en avbruten load multiple eller store multiple instruktion.

ICI:

Då ett avbrott inträffar under exekvering av någon av instruktionerna LDM STM, PUSH, POP, VLDM, VSTM, VPOP, eller VPOP:

- Stoppas instruktionen temporärt
- Skriver värdet för operationens nästa register i EPSR [15:12].

Efter att ha betjänat avbrottet:

- Fullföljer instruktionen med början på det register som anges av EPSR[15:12]
Bitarna [26:25,11:10] är 0 om processorn är i ICI-tillstånd.

IT:

Ett IT-block utgörs av upp till fyra villkorligt exekverbara instruktioner. Se instruktionslistan för beskrivning av IT-instruktionen.

T: Thumb state

Cortex M4 stödjer enbart exekvering av instruktioner i Thumb-tillstånd. Följande instruktioner kan potentiellt ändra detta tillstånd eftersom de påverkar T-biten.

:

- Instruktionerna BLX, BX och POP{PC}
- Återställning av av xPSR vid återgång från undantag
- Bit[0] i adressen hos en vektor i avbrottstabellen

Försök att exekvera en instruktion då T är 0 resulterar i undantag (fault) eller att processorn stannar (lockup).

Läsning från EPSR med instruktionen MSR returnerar alltid 0. Försök att skriva ignoreras. EPSR kan undersökas i en hanteringsrutin genom att EPSR extraheras från det PSR som lagrats på stacken vid undantag/avbrottet.

CONTROL:

FLOATING POINT ENABLE

F=0: Flyttalsräknaren används ej

F=1: Flyttalsräknare aktiverad

SPSEL: Aktiv stackpekare

S=0: MSP är aktiv stackpekare

S=1: PSP är aktiv stackpekare

I Handler mode, läses alltid denna bit som 0. Processorn återställer automatiskt rätt bit vid återgång från undantag/avbrott.

nPRIV: Nivå då processorn är i Thread mode.

P=0: Priviligierad, alla instruktioner tillgängliga.

P=1: Icke privilegierad, försök att exekvera privilegierad instruktion resulterar i undantag.

BASEPRI[7:4] Prioritetsmask för avbrott

0x00: ingen betydelse

Ett värde skilt från noll i dessa bitar anger den basprioritet som gäller. Processorn accepterar inga avbrott med ett prioritetvärde som är större eller lika med värdet i detta register. Observera att högsta prioritet anges med prioritetvärdet 0, lägsta prioritet anges med värdet 0xF0.

PRIMASK:

0: Ingen effekt

1: Förhindrar aktivering av undantag/avbrott med konfigurerbar prioritet, dvs *maskerar* avbrott..

FAULTMASK:

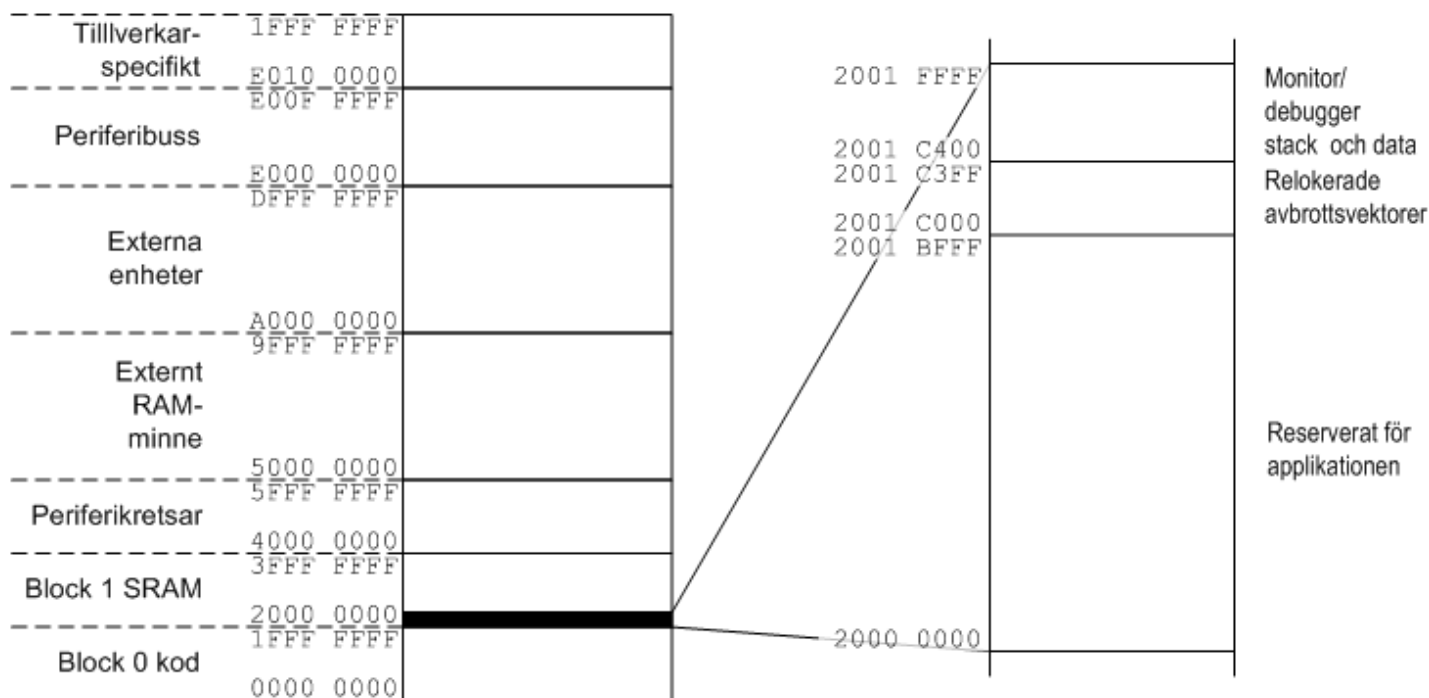
0: Ingen effekt

1: Förhindrar aktivering av alla undantag utom NMI.

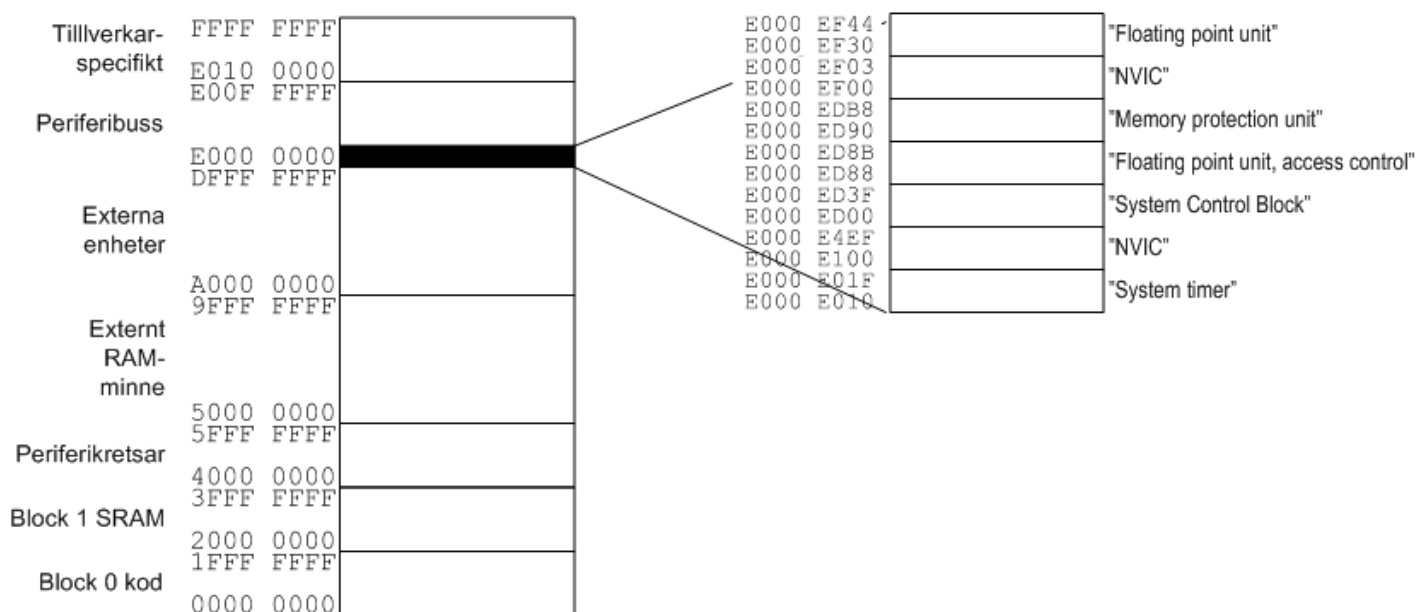
ADRESSERINGSSÄTT

Namn	Syntax	Exempel	RTN
Register direct	Rx	MOV R0,R1	R0←R1
Direct	Symbol	LDR R0,symbol	R0←M(symbol)
Immediate	#const	MOV R0,#0x15	R0←0x15
Register indirect	[Rx]	LDR R0,[R1]	R0←M(R1)
.. with offset	[Rx,#offset]	LDR R0,[R1,#4]	R0←M(R1+4)
.. with pre-increment	[Rx,#offset]!	LDR R0,[R1,#4]!	R1←R1+4, R0←M(R1)
.. with post-increment	[Rx],#offset	LDR R0,[R1],#4	R0←M(R1), R1←R1+4
.. with register index	[Rx,Ri]	LDR R0,[R1,R2]	R0←M(R1+R2)
.. with scaled index	[Rx,Ri,shift]	LDR R0,[R1,R2,LSL #2]	R0←M(R1+(R2<<2))

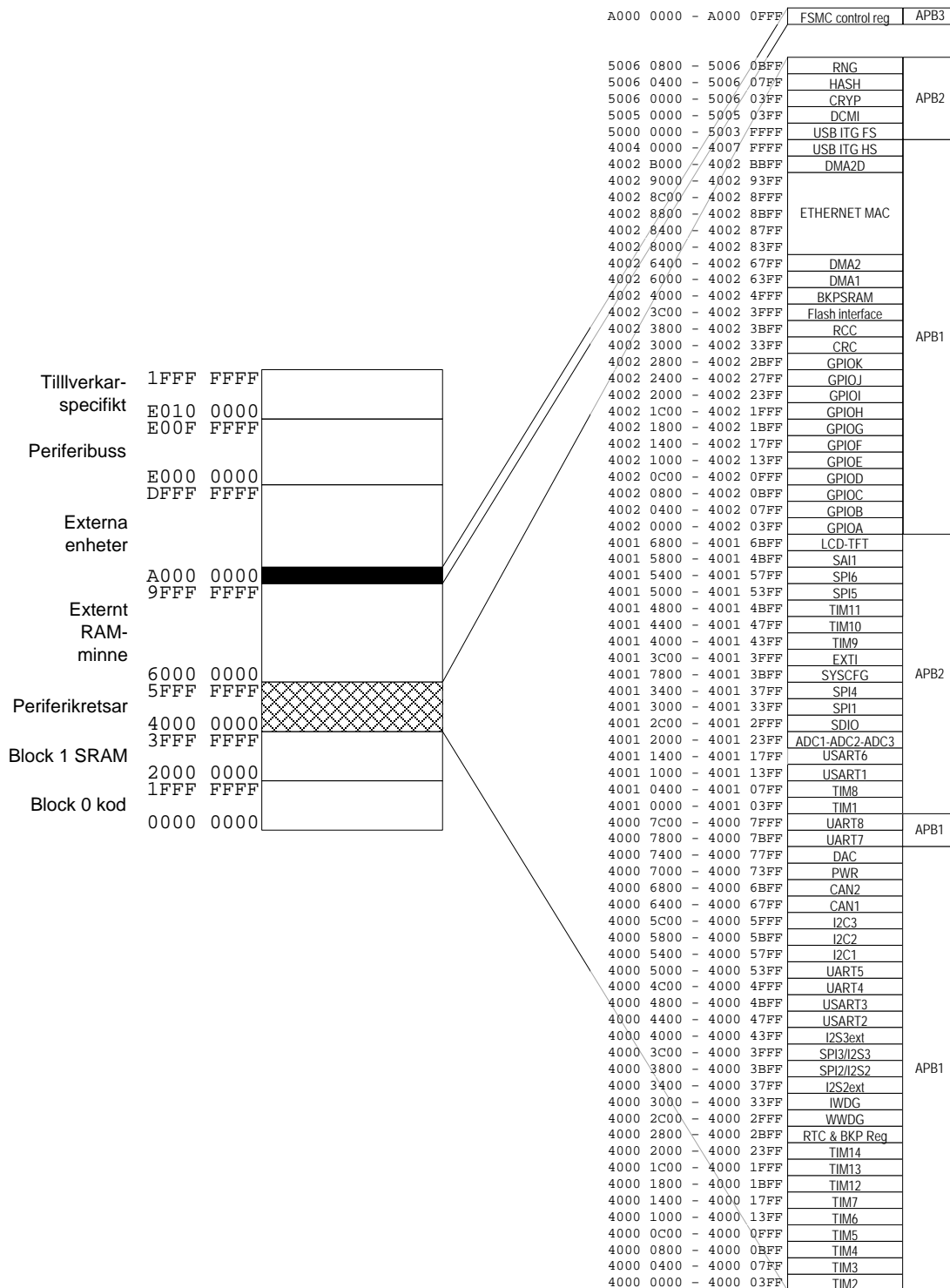
Minnesdisposition MD407



Minnesdisposition Systemmoduler för ST32F407



Minnesdisposition IO för ST32F407



Instruktionsuppsättning

Thumb® 16-bit Instruction Set Quick Reference Card

This card lists all Thumb instructions available on Thumb-capable processors earlier than ARM®v6T2. In addition, it lists all Thumb-2 16-bit instructions. The instructions shown on this card are all 16-bit in Thumb-2, except where noted otherwise. All registers are Lo (R0-R7) except where specified. Hi registers are R8-R15.

Key to Tables

§	See Table ARM architecture versions .	<!oreglist+LR>	A comma-separated list of Lo registers, plus the LR, enclosed in braces, { and }.
<!oreglist>	A comma-separated list of Lo registers, enclosed in braces, { and }.	<!oreglist+PC>	A comma-separated list of Lo registers, plus the PC, enclosed in braces, { and }.

Operation	§	Assembler	Updates	Action	Notes	
Move	Immediate	MOV _S Rd, #<imm>	N Z	Rd := imm	imm range 0-255.	
	Lo to Lo	MOV _S Rd, Rm	N Z	Rd := Rm	Synonym of LSL _S Rd, Rm, #0	
Add	Hi to Lo, Lo to Hi, Hi to Hi	MOV Rd, Rm		Rd := Rm	Not Lo to Lo.	
	Any to Any	MOV Rd, Rm		Rd := Rm	Any register to any register.	
	Immediate 3	ADD _S Rd, Rn, #<imm>	N Z C V	Rd := Rn + imm	imm range 0-7.	
	All registers Lo	ADD _S Rd, Rn, Rm	N Z C V	Rd := Rn + Rm	Not Lo to Lo.	
	Hi to Lo, Lo to Hi, Hi to Hi	ADD Rd, Rd, Rm	N Z C V	Rd := Rd + Rm	Any register to any register.	
	Any to Any	ADD Rd, Rd, Rm		Rd := Rd + Rm	imm range 0-255.	
	Immediate 8	ADD _S Rd, Rd, #<imm>	N Z C V	Rd := Rd + imm		
	With carry	ADC _S Rd, Rd, Rm	N Z C V	Rd := Rd + Rm + C-bit		
	Value to SP	ADD SP, SP, #<imm>	N Z C V	SP := SP + imm		
	Form address from SP	ADD Rd, SP, #<imm>	N Z C V	Rd := SP + imm		
Subtract	Form address from PC	ADR Rd, <label>		Rd := label	label range PC to PC+1020 (word-aligned).	
	Lo and Lo	SUB _S Rd, Rn, Rm	N Z C V	Rd := Rn - Rm		
	Immediate 3	SUB _S Rd, Rn, #<imm>	N Z C V	Rd := Rn - imm	imm range 0-7.	
	Immediate 8	SUB _S Rd, Rd, #<imm>	N Z C V	Rd := Rd - imm	imm range 0-255.	
	With carry	SBC _S Rd, Rd, Rm	N Z C V	Rd := Rd - Rm - NOT C-bit		
	Value from SP	SUB SP, SP, #<imm>	N Z C V	SP := SP - imm		
	Negate	RSB _S Rd, Rn, #0	N Z C V	Rd := -Rn	imm range 0-508 (word-aligned).	
	Multiply	MUL _S Rd, Rm, Rd	N Z * *	Rd := Rm * Rd	Synonym: NEGS Rd, Rn * C and V flags unpredictable in §4T, unchanged in §5T and above	
	Compare	Negative	CMP Rn, Rm	N Z C V	update APSR flags on Rn - Rm	Can be Lo to Lo, Lo to Hi, Hi to Lo, or Hi to Hi.
		Immediate	CMN Rn, Rm	N Z C V	update APSR flags on Rn + Rm	
		CMP Rn, #<imm>	N Z C V	update APSR flags on Rn - imm	imm range 0-255.	
Logical	AND	AND _S Rd, Rd, Rm	N Z	Rd := Rd AND Rm		
	Exclusive OR	EOR _S Rd, Rd, Rm	N Z	Rd := Rd EOR Rm		
	OR	ORR _S Rd, Rd, Rm	N Z	Rd := Rd OR Rm		
	Bit clear	BIC _S Rd, Rd, Rm	N Z	Rd := Rd AND NOT Rm		
	Move NOT	MVNS Rd, Rd, Rm	N Z	Rd := NOT Rm		
	Test bits	TST Rn, Rm	N Z	update APSR flags on Rn AND Rm		
	Shift/rotate	Logical shift left	LSL _S Rd, Rm, #<shift>	N Z C*	Rd := Rm << shift	Allowed shifts 0-31. * C flag unaffected if shift is 0.
Logical shift right		LSR _S Rd, Rd, Rn	N Z C*	Rd := Rd << Rn[7:0]	* C flag unaffected if Rn[7:0] is 0.	
Arithmetic shift right		ASR _S Rd, Rm, #<shift>	N Z C*	Rd := Rm >> shift	Allowed shifts 1-32.	
		ASR _S Rd, Rd, Rn	N Z C*	Rd := Rd >> Rn[7:0]	* C flag unaffected if Rn[7:0] is 0.	
		ASR _S Rd, Rm, #<shift>	N Z C*	Rd := Rm ASR shift	Allowed shifts 1-32.	
		ASR _S Rd, Rd, Rn	N Z C*	Rd := Rd ASR Rn[7:0]	* C flag unaffected if Rn[7:0] is 0.	
Rotate right		ROR _S Rd, Rd, Rn	N Z C*	Rd := Rd ROR Rn[7:0]	* C flag unaffected if Rn[7:0] is 0.	

Thumb 16-bit Instruction Set Quick Reference Card

Operation	§	Assembler	Action	Notes
Load	with immediate offset, word	LDR Rd, [Rn, #<imm>]	Rd := [Rn + imm]	imm range 0-124, multiple of 4.
	halfword	LDRH Rd, [Rn, #<imm>]	Rd := ZeroExtend([Rn + imm][15:0])	Clears bits 31:16, imm range 0-62, even.
	byte	LDRB Rd, [Rn, #<imm>]	Rd := ZeroExtend([Rn + imm][7:0])	Clears bits 31:8, imm range 0-31.
	with register offset, word	LDR Rd, [Rn, Rm]	Rd := [Rn + Rm]	Clears bits 31:16
	halfword	LDRH Rd, [Rn, Rm]	Rd := ZeroExtend([Rn + Rm][15:0])	Sets bits 31:16 to bit 15
	signed halfword	LDRSH Rd, [Rn, Rm]	Rd := SignExtend([Rn + Rm][15:0])	Clears bits 31:8
	byte	LDRB Rd, [Rn, Rm]	Rd := ZeroExtend([Rn + Rm][7:0])	Sets bits 31:8 to bit 7
	signed byte	LDRSB Rd, [Rn, Rm]	Rd := SignExtend([Rn + Rm][7:0])	label range PC to PC+1020 (word-aligned).
	PC-relative	LDR Rd, <label>	Rd := [label]	imm range 0-1020, multiple of 4.
	SP-relative	LDR Rd, [SP, #<imm>]	Rd := [SP + imm]	Always updates base register, Increment After.
Store	Multiple, not including base	LDM Rn!, <loreglist>	Loads list of registers (not including Rn)	Never updates base register, Increment After.
	Multiple, including base	LDM Rn, <loreglist>	Loads list of registers (including Rn)	
	with immediate offset, word	STR Rd, [Rn, #<imm>]	[Rn + imm] := Rd	imm range 0-124, multiple of 4.
	halfword	STRH Rd, [Rn, #<imm>]	[Rn + imm][15:0] := Rd[15:0]	Ignores Rd[31:16], imm range 0-62, even.
	byte	STRB Rd, [Rn, #<imm>]	[Rn + imm][7:0] := Rd[7:0]	Ignores Rd[31:8], imm range 0-31.
	with register offset, word	STR Rd, [Rn, Rm]	[Rn + Rm] := Rd	Ignores Rd[31:16]
	halfword	STRH Rd, [Rn, Rm]	[Rn + Rm][15:0] := Rd[15:0]	Ignores Rd[31:8]
	byte	STRB Rd, [Rn, Rm]	[Rn + Rm][7:0] := Rd[7:0]	imm range 0-1020, multiple of 4.
	SP-relative, word	STR Rd, [SP, #<imm>]	[SP + imm] := Rd	Always updates base register, Increment After.
	Multiple	STM Rn!, <loreglist>	Stores list of registers	
Push	Push	PUSH <loreglist>	Push registers onto full descending stack	
	Push with link	PUSH <loreglist>+LR	Push LR and registers onto full descending stack	
Pop	Pop	POP <loreglist>	Pop registers from full descending stack	
	Pop and return	POP <loreglist>+PC	Pop registers, branch to address loaded to PC	
	Pop and return with exchange	POP <loreglist>+PC>	Pop, branch, and change to ARM state if address[0] = 0	
If-Then	If-Then	T2 IT(pattern) {cond}	Makes up to four following instructions conditional, according to pattern. pattern is a string of up to three letters. Each letter can be T (Then) or E (Else).	The first instruction after IT has condition cond. The following instructions have condition cond if the corresponding letter is T, or the inverse of cond if the corresponding letter is E. See Table Condition Field .
	Conditional branch	B{cond} <label>	If {cond} then PC := label	label must be within -252 to +258 bytes of current instruction. See Table Condition Field .
Branch	Compare, branch if (non) zero	T2 CB(N)Z Rn, <label>	If Rn (== !=) 0 then PC := label	label must be within +4 to +130 bytes of current instruction.
	Unconditional branch	B <label>	PC := label	label must be within ±2KB of current instruction.
	Long branch with link	BL <label>	LR := address of next instruction, PC := label	This is a 32-bit instruction.
	Branch and exchange	BX Rm	PC := Rm AND 0xFFFFFFFF	label must be within ±4MB of current instruction (T2: ±16MB).
	Branch with link and exchange	BLX <label>	LR := address of next instruction, PC := label	Change to ARM state if Rm[0] = 0.
Branch with link and exchange	Branch with link and exchange	BLX Rm	LR := address of next instruction, PC := label	This is a 32-bit instruction.
	Branch with link and exchange	BLX Rm	LR := address of next instruction, PC := Rm AND 0xFFFFFFFF	label must be within ±4MB of current instruction (T2: ±16MB).
Extend	Signed, halfword to word	6 SXTH Rd, Rm	Rd[31:0] := SignExtend(Rm[15:0])	
	Signed, byte to word	6 SXTB Rd, Rm	Rd[31:0] := SignExtend(Rm[7:0])	
	Unsigned, halfword to word	6 UXTB Rd, Rm	Rd[31:0] := ZeroExtend(Rm[15:0])	
	Unsigned, byte to word	6 UXTB Rd, Rm	Rd[31:0] := ZeroExtend(Rm[7:0])	
	Bytes in word	6 REV Rd, Rm	Rd[31:24] := Rm[7:0], Rd[23:16] := Rm[15:8], Rd[15:8] := Rm[23:16], Rd[7:0] := Rm[31:24]	
	Bytes in both halfwords	6 REV16 Rd, Rm	Rd[15:8] := Rm[7:0], Rd[7:0] := Rm[15:8], Rd[31:24] := Rm[23:16], Rd[23:16] := Rm[31:24]	
Reverse	Bytes in low halfword, sign extend	6 REVSH Rd, Rm	Rd[15:8] := Rm[7:0], Rd[7:0] := Rm[15:8], Rd[31:16] := Rm[7] * &FFFF	

Thumb 16-bit Instruction Set Quick Reference Card

Operation	\$	Assembler	Action	Notes
Processor state change				
Supervisor Call		SVC <immed_8>	Supervisor Call processor exception	8-bit immediate value encoded in instruction. Formerly SWI.
Change processor state	6	CPSID <iFlags>	Disable specified interrupts	
	6	CPSIE <iFlags>	Enable specified interrupts	
Set endianness	6	SETEND <endianness>	Sets endianness for loads and saves.	<endianness> can be BE (Big Endian) or LE (Little Endian).
Breakpoint	5T	BKPT <immed_8>	Prefetch abort or enter debug state	8-bit immediate value encoded in instruction.
No Op				
No operation		NOP	None, might not even consume any time.	Real NOP available in ARM v6K and above.
Hint				
Set event	T2	SEV	Signal event in multiprocessor system.	Executes as NOP in Thumb-2. Functionally available in ARM v7.
Wait for event	T2	WFE	Wait for event, IRQ, FIQ, Imprecise abort, or Debug entry request.	Executes as NOP in Thumb-2. Functionally available in ARM v7.
Wait for interrupt	T2	WFI	Wait for IRQ, FIQ, Imprecise abort, or Debug entry request.	Executes as NOP in Thumb-2. Functionally available in ARM v7.
Yield	T2	YIELD	Yield control to alternative thread.	Executes as NOP in Thumb-2. Functionally available in ARM v7.

Condition Field	Mnemonic	Description
	EQ	Equal
	NE	Not equal
	CS / HS	Carry Set / Unsigned higher or same
	CC / LO	Carry Clear / Unsigned lower
	MI	Negative
	PL	Positive or zero
	VS	Overflow
	VC	No overflow
	HI	Unsigned higher
	LS	Unsigned lower or same
	GE	Signed greater than or equal
	LT	Signed less than
	GT	Signed greater than
	LE	Signed less than or equal
	AL	Always. Do not use in B{cond}

In Thumb code for processors earlier than ARMv6T2, cond must not appear anywhere except in Conditional Branch (B{cond}) instructions.

In Thumb-2 code, cond can appear in any of these instructions (except CBZ, CBNZ, CPSID, CPSIE, IT, and SETEND).
The condition is encoded in a preceding IT instruction (except in the case of B{cond} instructions).
If IT instructions are explicitly provided in the Assembly language source file, the conditions in the instructions must match the corresponding IT instructions.

ARM architecture versions

4T	All Thumb versions of ARM v4 and above.
5T	All Thumb versions of ARM v5 and above.
6	All Thumb versions of ARM v6 and above.
T2	All Thumb-2 versions of ARM v6 and above.

Proprietary Notice

Words and logos marked with [®] or [™] are registered trademarks or trademarks of ARM Limited in the EU and other countries, except as otherwise stated below in this proprietary notice. Other brands and names mentioned herein may be the trademarks of their respective owners.

Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder.

The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by ARM in good faith. However, all warranties implied or expressed, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This reference card is intended only to assist the reader in the use of the product. ARM Ltd shall not be liable for any loss or damage arising from the use of any information in this reference card, or any error or omission in such information, or any incorrect use of the product.

Document Number

ARM QRC 0006E

Change Log

Issue	Date	Change
A	Nov 2004	First Release
B	May 2005	RVCT 2.2 SPI
C	March 2006	RVCT 3.0
D	March 2007	RVCT 3.1
E	Sept 2008	RVCT 4.0

AVBROTTSVEKTORER

num	pri					
				Reserved for initial stack pointer	0x0000 0000	
	-3	fixed	Reset	Reset	0x0000 0004	
	-2	fixed	NMI	Non maskable interrupt. The RCC Clock Security System (CSS) is inked to the NMI vector.	0x0000 0008	
	-1	fixed	HardFault	All class of fault	0x0000 000C	
	0	settable	MemManage	Memory management	0x0000 0010	
	1	settable	BusFault	Pre-fetch fault, memory access fault	0x0000 0014	
	2	settable	UsageFault	Undefined instruction or illegal state	0x0000 0018	
	.	.	.	Reserved	0x0000 001C - 0x0000 002B	
	3	settable	SVCall	System service call via SWI instruction	0x0000 002C	
	4	settable	Debug Monitor	Debug Monitor	0x0000 0030	
	-	-	-	Reserved	0x0000 0034	
	5	settable	PendSV	Pendable request for system service	0x0000 0038	
	6	settable	SysTick	System tick timer	0x0000 003C	
	0	7	settable	WWDG	Window Watchdog interrupt	0x0000 0040
	1	8	settable	PVD	PVD through EXTI line detection interrupt	0x0000 0044
	2	9	settable	TAMP_STAMP	Tamper and TimeStamp interrupts through the EXTI line	0x0000 0048
	3	10	settable	RTC_WKUP	RTC Wakeup interrupt through the EXTI line	0x0000 004C
	4	11	settable	FLASH	Flash global interrupt	0x0000 0050
	5	12	settable	RCC	RCC global interrupt	0x0000 0054
	6	13	settable	EXTI0	EXTI Line0 interrupt	0x0000 0058
	7	14	settable	EXTI1	EXTI Line1 interrupt	0x0000 005C
	8	15	settable	EXTI2	EXTI Line2 interrupt	0x0000 0060
	9	16	settable	EXTI3	EXTI Line3 interrupt	0x0000 0064
	10	17	settable	EXTI4	EXTI Line4 interrupt	0x0000 0068
	11	18	settable	DMA1_Stream0	DMA1 Stream0 global interrupt	0x0000 006C
	12	19	settable	DMA1_Stream1	DMA1 Stream1 global interrupt	0x0000 0070
	13	20	settable	DMA1_Stream2	DMA1 Stream2 global interrupt	0x0000 0074
	14	21	settable	DMA1_Stream3	DMA1 Stream3 global interrupt	0x0000 0078
	15	22	settable	DMA1_Stream4	DMA1 Stream4 global interrupt	0x0000 007C
	16	23	settable	DMA1_Stream5	DMA1 Stream5 global interrupt	0x0000 0080
	17	24	settable	DMA1_Stream6	DMA1 Stream6 global interrupt	0x0000 0084
	18	25	settable	ADC	ADC1, ADC2 and ADC3 global interrupts	0x0000 0088
	19	26	settable	CAN1_TX	CAN1 TX interrupts	0x0000 008C
	20	27	settable	CAN1_RX0	CAN1 RX0 interrupts	0x0000 0090
	21	28	settable	CAN1_RX1	CAN1 RX1 interrupt	0x0000 0094
	22	29	settable	CAN1_SCE	CAN1 SCE interrupt	0x0000 0098
	23	30	settable	EXTI9_5 EXTI	Line[9:5] interrupts	0x0000 009C
	24	31	settable	TIM1_BRK_TIM9	TIM1 Break interrupt and TIM9 global interrupt	0x0000 00A0
	25	32	settable	TIM1_UP_TIM10	TIM1 Update interrupt and TIM10 global interrupt	0x0000 00A4
	26	33	settable	TIM1_TRG_COM_TIM11	TIM1 Trigger and Commutation interrupts and TIM11 global interrupt	0x0000 00A8
	27	34	settable	TIM1_CC	TIM1 Capture Compare interrupt	0x0000 00AC
	28	35	settable	TIM2	TIM2 global interrupt	0x0000 00B0
	29	36	settable	TIM3	TIM3 global interrupt	0x0000 00B4
	30	37	settable	TIM4	TIM4 global interrupt	0x0000 00B8
	31	38	settable	I2C1_EV	I2C1 event interrupt	0x0000 00BC
	32	39	settable	I2C1_ER	I2C1 error interrupt	0x0000 00C0
	33	40	settable	I2C2_EV	I2C2 event interrupt	0x0000 00C4
	34	41	settable	I2C2_ER	I2C2 error interrupt	0x0000 00C8
	35	42	settable	SPI1	SPI1 global interrupt	0x0000 00CC
	36	43	settable	SPI2	SPI2 global interrupt	0x0000 00D0
	37	44	settable	USART1	USART1 global interrupt	0x0000 00D4
	38	45	settable	USART2	USART2 global interrupt	0x0000 00D8
	39	46	settable	USART3	USART3 global interrupt	0x0000 00DC
	40	47	settable	EXTI15_10	EXTI Line[15:10] interrupts	0x0000 00E0
	41	48	settable	RTC_Alarm	RTC Alarms (A and B) through EXTI line interrupt	0x0000 00E4
	42	49	settable	OTG_FS	WKUP USB On-The-Go FS Wakeup through EXTI line interrupt	0x0000 00E8
	43	50	settable	TIM8_BRK_TIM12	TIM8 Break interrupt and TIM12 global interrupt	0x0000 00EC
	44	51	settable	TIM8_UP_TIM13	TIM8 Update interrupt and TIM13 global interrupt	0x0000 00F0
	45	52	settable	TIM8_TRG_COM_TIM14	TIM8 Trigger and Commutation interrupts and TIM14 global interrupt	0x0000 00F4
	46	53	settable	TIM8_CC	TIM8 Capture Compare interrupt	0x0000 00F8
	47	54	settable	DMA1_Stream7	DMA1 Stream7 global interrupt	0x0000 00FC
	48	55	settable	FSMC	FSMC global interrupt	0x0000 0100
	49	56	settable	SDIO	SDIO global interrupt	0x0000 0104
	50	57	settable	TIM5	TIM5 global interrupt	0x0000 0108
	51	58	settable	SPI3	SPI3 global interrupt	0x0000 010C
	52	59	settable	UART4	UART4 global interrupt	0x0000 0110
	53	60	settable	UART5	UART5 global interrupt	0x0000 0114
	54	61	settable	TIM6_DAC	TIM6 global interrupt, DAC1 and DAC2 underrun error interrupts	0x0000 0118
	55	62	settable	TIM7	TIM7 global interrupt	0x0000 011C
	56	63	settable	DMA2_Stream0	DMA2 Stream0 global interrupt	0x0000 0120

57	64	settable	DMA2_Stream1	DMA2 Stream1 global interrupt	0x0000 0124
58	65	settable	DMA2_Stream2	DMA2 Stream2 global interrupt	0x0000 0128
59	66	settable	DMA2_Stream3	DMA2 Stream3 global interrupt	0x0000 012C
60	67	settable	DMA2_Stream4	DMA2 Stream4 global interrupt	0x0000 0130
61	68	settable	ETH	Ethernet global interrupt	0x0000 0134
62	69	settable	ETH_WKUP	Ethernet Wakeup through EXTI line interrupt	0x0000 0138
63	70	settable	CAN2_TX	CAN2 TX interrupts	0x0000 013C
64	71	settable	CAN2_RX0	CAN2 RX0 interrupts	0x0000 0140
65	72	settable	CAN2_RX1	CAN2 RX1 interrupt	0x0000 0144
66	73	settable	CAN2_SCE	CAN2 SCE interrupt	0x0000 0148
67	74	settable	OTG_FS	USB On The Go FS global interrupt	0x0000 014C
68	75	settable	DMA2_Stream5	DMA2 Stream5 global interrupt	0x0000 0150
69	76	settable	DMA2_Stream6	DMA2 Stream6 global interrupt	0x0000 0154
70	77	settable	DMA2_Stream7	DMA2 Stream7 global interrupt	0x0000 0158
71	78	settable	USART6	USART6 global interrupt	0x0000 015C
72	79	settable	I2C3_EV	I2C3 event interrupt	0x0000 0160
73	80	settable	I2C3_ER	I2C3 error interrupt	0x0000 0164
74	81	settable	OTG_HS_EP1_OUT	USB On The Go HS End Point 1 Out global interrupt	0x0000 0168
75	82	settable	OTG_HS_EP1_IN	USB On The Go HS End Point 1 In global interrupt	0x0000 016C
76	83	settable	OTG_HS_WKUP	USB On The Go HS Wakeup through EXTI interrupt	0x0000 0170
77	84	settable	OTG_HS	USB On The Go HS global interrupt	0x0000 0174
78	85	settable	DCMI	DCMI global interrupt	0x0000 0178
79	86	settable	CRYP	CRYP crypto global interrupt	0x0000 017C
80	87	settable	HASH_RNG	Hash and Rng global interrupt	0x0000 0180
81	88	settable	FPU	FPU global interrupt	0x0000 0184

GCC KOMPILATORKONVENTIONER

Register	Synonym	Special	Role in the procedure call standard
r15		PC	The Program Counter.
r14		LR	The Link Register.
r13		SP	The Stack Pointer.
r12		IP	The Intra-Procedure-call scratch register.
r11	v8		Variable-register 8.
r10	v7		Variable-register 7.
r9		v6 SB TR	Platform register. The meaning of this register is defined by the platform standard.
r8	v5		Variable-register 5.
r7	v4		Variable register 4.
r6	v3		Variable register 3.
r5	v2		Variable register 2.
r4	v1		Variable register 1.
r3	a4		Argument / scratch register 4.
r2	a3		Argument / scratch register 3.
r1	a2		Argument / result / scratch register 2.
r0	a1		Argument / result / scratch register 1.

PERIFERIENHETER, "CORE PERIPHERALS" STM32

SysTick (0xE000E010-0xE000E01F) System Timer

offset	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	Register	
0																rw															rw	rw	rw	STK_CTRL
4									rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	STK_LOAD	
8									rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	STK_VAL	
0xC	r	r							r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	STK_CALIB		

STK_CTRL (0xE000E010) Status och styrregister

offset	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	Register	
0																rw															rw	rw	rw	STK_CTRL

Bit 16: (COUNTFLAG):

Biten är 1 om räknaren räknat ned till 0. Biten nollställs då registret läses.

Bit 2: (CLKSOURCE) biten är 1 efter RESET:

- 0: Systemklocka/8
- 1: Systemklocka

Bit 1: (TICKINT): Aktivera avbrott

- 0: Inget avbrott genereras.
- 1: Då räknaren slår om till 0 genereras SysTick avbrott.

STK_LOAD (0xE000E014) Räknarintervall

offset	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	Register
4									rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	STK_LOAD

Bits 23:0 Värde för räknarintervall

Registret håller räknarintervallens startvärde. Värdet kan vara i intervallet 0x00000001-0x00FFFFFF.

Startvärdet 0 är möjligt men meningslöst eftersom slutet av räknarintervallet detekteras av att räknaren slår om från 1 till 0. För att generera N cykler fördröjning ska därför värdet i STK_VAL sättas till $N-1$.

STK_VAL (0xE000E018) Räknarvärde

offset	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	Register
8									rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	STK_VAL

Bits 23:0 Aktuellt räknarvärde

Läsning av registret returnerar räknarens aktuella värde. En skrivning till registret, oavsett värde, nollställer registret såväl som COUNTFLAG i statusregistret.

STK_CALIB (0xE000E00C) Kaibreringsregister

offset	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	Register
0xC	r	r							r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	STK_CALIB	

Registret är avsett för kalibreringsändamål och innehåller implementeringsspecifika detaljer. Vi behandlar inte dessa här.

NVIC (0xE00E100-0xE00E4EF) Nested vectored interrupt controller

För varje avbrott kontrollerat av NVIC finns följande funktioner:

- *Interrupt Set Enable Registers*, NVIC_ISERx
- *Interrupt Clear Enable Registers*, NVIC_ICERx
- *Interrupt Set Pending Registers*, NVIC_ISEPRx
- *Interrupt Clear Pending Registers*, NVIC_ICPRx
- *Interrupt Active Bit Registers*, NVIC_IABRx
- *Interrupt Priority Registers*, NVIC_IPRx
- *Software Interrupt Trigger Register*, NVIC_STIR

offset	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	mnemonic
0x000	SETENA[31:0]																																NVIC_ISER0
0x004	SETENA[63:32]																																NVIC_ISER1
0x008	Reserverat																SETENA[80:64]																NVIC_ISER2
0x080	CLRENA[31:0]																																NVIC_CER0
0x084	CLRENA [63:32]																																NVIC_CER1
0x088	Reserverat																CLRENA [80:64]																NVIC_CER2
0x100	SETPEND[31:0]																																NVIC_ISPR0
0x104	SETPEND [63:32]																																NVIC_ISPR1
0x108	Reserverat																SETPEND [80:64]																NVIC_ISPR2
0x180	CLRPEND[31:0]																																NVIC_ICPR0
0x184	CLRPEND [63:32]																																NVIC_ICPR1
0x188	Reserverat																CLRPEND [80:64]																NVIC_ICPR2
0x200	ACTIVE[31:0]																																NVIC_IABR0
0x204	ACTIVE [63:32]																																NVIC_IABR1
0x208	Reserverat																ACTIVE [80:64]																NVIC_IABR2
0x300	IP[3]								IP[2]								IP[1]								IP[0]								NVIC_IPR0
...
0x320	Reserverat																								IP[80]								NVIC_IPR20

NVIC_ISRx: SETENA[80..0] Interrupt set-enable bit:

Skrivning:

- 0: ingen effekt
- 1: möjliggör avbrott

Läsning:

- 0: avbrott avstängt
- 1: avbrott möjligt

NVIC_ICERx: CLRENA[80..0] Interrupt clear-enable bit:

Skrivning:

- 0: ingen effekt
- 1: omöjliggör avbrott

Läsning:

- 0: avbrott avstängt
- 1: avbrott möjligt

NVIC_ISEPRx: SETPEND[80..0] Interrupt set pending bit:

Skrivning:

- 0: ingen effekt
- 1: ändrar avbrottstatus till "avvaktande" (pending)

Läsning:

- 0: avbrottstatus är inte "avvaktande"
- 1: avbrottstatus är "avvaktande"

NVIC_ICPRx: CLRPEND[80..0] Interrupt clear pending bit:

Skrivning:

- 0: ingen effekt
- 1: avlägsnar avbrottstatus "avvaktande" (pending)

Läsning:

- 0: avbrottstatus är inte "avvaktande"
- 1: avbrottstatus är "avvaktande"

NVIC_IABRx: ACTIVE[80..0] Interrupt active bit:

Läsning:

- 0: avbrottstatus är inte "aktivt"
- 1: avbrottstatus är "aktivt"

NVIC_IPRx: Interrupt priority

Varje prioritetsfält kan ha ett prioritetsvärde , 0-255. Ju lägre värde, desto högre prioritet för motsvarande avbrott. Processorn implementerar bara bitar [7: 4] för varje fält, bitar [3: 0] läses som noll och ignoreras vid skrivning.

RCC, RESET AND CLOCK CONTROL, STM32F407

offset	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	Register
0																																	RCC_CR
4																																	RCC_PLLCFGR
8																																	RCC_CFGR
0xC																																	RCC_CIR
0x10																																	RCC_AHB1RSTR
0x14																																	RCC_AHB2RSTR
0x18																																	RCC_AHB3RSTR
0x1C																																	
0x20																																	RCC_ABP1RSTR
0x24																																	RCC_ABP2RSTR
0x28																																	
0x2C																																	
0x30																																	RCC_AHB1ENR
0x34																																	RCC_AHB2ENR
0x38																																	RCC_AHB3ENR
0x3C																																	
0x40																																	RCC_APB1ENR
0x44																																	RCC_APB2ENR
0x48																																	
0x4C																																	
0x50																																	RCC_AHB1LPENR
0x54																																	RCC_AHB2LPENR
0x58																																	RCC_AHB3LPENR
0x5C																																	
0x60																																	RCC_APB1LPENR
0x64																																	RCC_APB2LPENR
0x68																																	
0x6C																																	
0x70																																	RCC_BDCR
0x74																																	RCC_CSR
0x78																																	
0x7C																																	
0x80																																	RCC_SSCGR
0x84																																	RCC_PLLI2SCFGR

GPIO, General Purpose I/Os

offset	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	Register
0																																	MODER
4																																	OTYPER
8																																	OSPEEDR
0xC																																	PUPDR
0x10																																	IDR
0x14																																	ODR
0x18																																	BSRR
0x1C																																	LCKR
0x20																																	AFRL
0x24																																	AFRH

GPIO Port mode register (MODER)

offset	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	mnemonic
0	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	MODER
	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	
				3						2						1						0											

För varje portpinne används 2 bitar i MODER för att konfigurera pinnen enligt:

- 00: ingång
- 01: utgång
- 10: alternativ funktion
- 11: analog

Registret har organiserats så att bitar 31,30 konfigurerar portpinne 15, bitar 29,28 konfigurerar portpinne 14 osv.

GPIO Output TYPE Register (OTYPER)

offset	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	mnemonic															
0x04	RESERVERADE																r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	OTYPER
																	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	
																	05						04																									

Bitar 31:16 är reserverade och ska inte ändras. För varje annan portpinne används 1 bit för att konfigurera pinnen enligt:

- 0: push-pull
- 1: open drain

GPIO Output SPEED Register (OSPEEDR)

offset	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	mnemonic		
0x08	r	w	r	w	r	w	r	w	r	w	r	w	r	w	r	w	r	w	r	w	r	w	r	w	r	w	r	w	r	w	r	w	r	w	OSPEEDR
	0x0B			0x0A			0x09			0x08																									

Registret används för att kontrollera uppdateringsfrekvensen för en portpinne. Exakta frekvenser anges i processorns datablad.

- 00: low speed
- 01: medium speed
- 10: fast speed
- 11: high speed

GPIO Input Data Register (IDR)

offset	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	mnemonic															
0x10	RESERVERADE																r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	IDR
																	0x11						0x10																									

Bitar 16 tom 31 används inte och ska hållas vid sitt RESET-värde, dvs 0. Bitar 0 t.o.m 15 avspeglar portens nivåer då pinnarna är konfigurerade som ingångar.

GPIO Pull-Up/Pull-Down Register (PUPDR)

offset	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	mnemonic
0x0C	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	PUPDR
	0x0F								0x0E								0x0D								0x0C								

För varje portpinne används 2 bitar i PUPDR för att konfigurera pinnen enligt:

00: floating

01: pull-up

10: pull-down

11: reserverad

GPIO Output Data Register (ODR)

offset	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	mnemonic														
0x14	RESERVERADE																r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	ODR
																	0x15								0x14																						

Bitar 16 tom 31 används inte och ska hållas vid sitt RESET-värde, dvs 0. Bitar 0 t.o.m. 15 sätter portens nivåer då pinnarna är konfigurerade som utgångar. Bitarna är både skriv- och läsbara, vid läsning ger biten det senast skrivna värdet till samma bit.

SYSCFG

offset	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	Register
0																																	SYSCFG_MEMRMP
4																																	SYSCFG_PMC
8																																	SYSCFG_EXTICR1
0xC																																	SYSCFG_EXTICR2
0x10																																	SYSCFG_EXTICR3
0x14																																	SYSCFG_EXTICR4
0x20																																	SYSCFG_CMPCR

SYSCFG_MEMRMP (0x40013800) Memory remap register

Registret används för att konfigurera adressutrymmet från adress 0 och uppåt. Normalt sett används de båda BOOT-pinnarna på kretsen för detta men med detta register kan det även göras i mjukvara. För MD407 gäller att FLASH-minnet, med startadress 0x08000000 dubbelavbildas från adress 0 och uppåt.

SYSCFG_PMC (0x40013804) Peripheral mode configuration register

offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Register
4									R_SEL								SYSCFG_PMC
									r/w					r/w	r/w	r/w	

Används för att konfigurera de fysiska egenskaperna hos Ethernet-gränssnittet (R_SEL) och välja funktion hos AD-omvandlare. Se tillverkarens anvisningar.

SYSCFG_EXTICR1 (0x40013808) External interrupt configuration register 1

offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Register
8	EXTI3[3:0]				EXTI2[3:0]				EXTI1[3:0]				EXTI0[3:0]				SYSCFG_EXTICR1
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

Bitarna i alla fyra CR-registren har samma betydelse, se CR4 nedan

SYSCFG_EXTICR2 (0x4001380C) External interrupt configuration register 2

offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Register
8	EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]				SYSCFG_EXTICR2
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

SYSCFG_EXTICR3 (0x40013810) External interrupt configuration register 3

offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Register
0x10	EXTI11[3:0]				EXTI10[3:0]				EXTI9[3:0]				EXTI8[3:0]				SYSCFG_EXTICR3
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

SYSCFG_EXTICR4 (0x40013814) External interrupt configuration register 4

offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Register
0x10	EXTI15[3:0]				EXTI14[3:0]				EXTI13[3:0]				EXTI12[3:0]				SYSCFG_EXTICR4
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

Bit 15:0 EXTIx[3:0]:

Dessa bitar bestämmer hur en IO-pinne dirigeras till någon av de 16 avbrottslinorna EXTI0..EXTI15 genom att motsvarande fält skrivs med fyra bitar. Undantag: PK[15:8] används ej.

0000: PA[x]	0001: PB[x]	0010: PC[x]	0011: PD[x]	0100: PE[x]	0101: PF[x]
0110: PG[x]	0111: PH[x]	1000: PI[x]	1001: PJ[x]	1010: PK[x]	

SYSCFG_CMPCR (0x40013820) Compensation cell control register

Används normalt inte men kan aktiveras för att öka stigtiden vid IO-frekvenser över 50MHz. Se tillverkarens anvisningar.

EXTI

offset	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	Register
0																																	EXTI_IMR
4																																	EXTI_EMR
8																																	EXTI_RTISR
0xC																																	EXTI_FTISR
0x10																																	EXTI_SWIER
0x14																																	EXTI_PR

EXTI_IMR (0x40013C00) Interrupt Mask Register

offset	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	mnemonic
0										rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	EXTI_IMR

Bit IMR[22..0]:

Avbrottsmask för avbrottslina x.

0: Avbrott är maskerat (deaktiverat)

1: Avbrott är aktiverat.

EXTI_EMR (0x40013C04) Event Mask Register

offset	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	mnemonic
4										rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	EXTI_EMR

Bit EMR[22..0]:

Eventmask för avbrottslina x.

0: Event är maskerat (deaktiverat)

1: Event är aktiverat.

Varje avbrottslina kan konfigureras för avbrott på positiv och/eller negativ flank.

EXTI_RTISR (0x40013C08) Rising Trigger Selection Register

offset	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	mnemonic
8										rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	EXTI_RTISR

Bit RTISR[22..0]:

Trigg på positiv flank för avbrottslina x.

0: Trigg på positiv flank är maskerat (deaktiverat)

1: Trigg på positiv flank är aktiverat.

EXTI_FTISR (0x40013C0C) Falling Trigger Selection Register

offset	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	mnemonic
0xC										rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	EXTI_FTISR

Bit FTISR[22..0]:

Trigg på negativ flank för avbrottslina x.

0: Trigg på negativ flank är maskerat (deaktiverat)

1: Trigg på negativ flank är aktiverat.

EXTI_SWIER (0x40013C10) Software Interrupt Event Register

offset	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	mnemonic
0x10										rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	EXTI_SWIER

Bit SWIER[22..0]:

Om avbrott är aktiverat för lina x, kan programvara aktivera detta genom att skriva '1' till motsvarande bit i detta register.

Denna bit återställs genom skrivning till EXTI_PR.

EXTI_PR (0x40013C14) Pending Register

offset	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	mnemonic
0x14										rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	EXTI_PR

Bit PR[22..0]:

Motsvarande bit sätts i detta register då ett triggvillkor är uppfyllt. Biten återställs genom att skrivas med '1'.

0: Trigg har uppträtt

1: Ingen Trigg.

USART, Universal synchronous asynchronous receiver transmitter

Översikt USART Register

offset	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	Register
0																																	SR
4																																	DR
8																																	BRR
0xC																																	CR1
0x10																																	CR2
0x14																																	CR3
0x18																																	GTPR

Åtkomst till registren kan ske som 16-bitars (HWORD) eller 32 bitars (WORD) ord.

USART_SR Statusregister

offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Register
4							CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE	USART_SR

Bit 9: CTS

Denna bit sätts av hårdvara om ETSC=1 när nivån på NCTS ingången ändras. Den återställs av programvara (genom att skriva 0 till biten). Ett avbrott genereras om CTSIE = 1 i registret UART_CR3.

0: Ingen förändring har skett på NCTS

1: En förändring har skett på NCTS

Bit 8: LBD

Denna bit sätts av hårdvara när LIN avbrott detekteras. Det återställs av programvara (genom att skriva 0 till biten). Ett avbrott genereras om LBDIE = 1 i registret USART_CR2.

0: LIN avbrott ej upptäckt

1: LIN avbrott upptäckt

Bit 7: TXE: "Transmit dataregister empty"

Denna bit sätts av hårdvara när innehållet av TDR registret har överförts till skiftregistret. Ett avbrott genereras om TXEIE bit = 1 i registret USART_CR1. Det återställs vid skrivning till USART_DR registret.

0: Dataregistrets sändardel är upptaget med en överföring.

1: Dataregistrets sändardel är klar att användas.

Bit 6 TC:

Denna bit sätts då sändningen av en ram är komplett och om TXE =1. Ett avbrott genereras om TCIE = 1 i USART_CR1 registret. TC nollställs av en läsning från USART_SR följt av en skrivning till USART_DR eller genom att "0" skrivs till biten.

0: Överföring pågår

1: Överföringen är klar

Bit 5 RXNE:

Denna bit sätts då innehållet i skiftregister RDR har överförts till USART_DR, dvs. ett nytt tecken har kommit. Ett avbrott genereras om RXNEIE = 1 i USART_CR1. Biten nollställs igen vid en läsning från USART_DR. Biten kan också återställas genom att skriva en nolla till den.

0: Inget nytt innehåll i USART_DR sedan senaste läsningen

1: Nytt innehåll finns i USART_DR.

Bit 4 IDLE:

Denna bit sätts av hårdvaran när en tom ram, indikerandes att serieledningen är ledig, upptäcks. Ett avbrott genereras om IDLEIE = 1 i USART_CR1. Det återställs av en läsning från USART_SR direkt följt av en läsning från USART_DR.

0: Ingen tom ram har detekterats

1: En tom ram har detekterats

Bit 3 ORE:

Denna bit sätts av hårdvaran om ett nytt tecken anländer samtidigt som det finns ett oläst tecken i dataregistret ("overrun error"). Ett avbrott genereras om RXNEIE = 1 i USART_CR1. Det återställs av en läsning från USART_SR följt av en läsning från USART_DR.

0: Inget förlorat tecken

1: Mottaget tecken är överskrivet (förlorat)

Bit 2 NF: Denna bit sätts av hårdvara när störningar i form av brus upptäcks i en mottagen ram. Biten återställs av en läsning från USART_SR följt av en läsning från USART_DR.

0: Ingen störning detekterad

1: Störning detekterad

Bit 1 FE:

Denna bit sätts av hårdvara när ett ramfel, oftast orsakat av tappad synkronisering, detekteras. Biten återställs av en läsning från USART_SR följt av en läsning från USART_DR.

0: Inget ramfel upptäcks

1: Ramfel eller BREAK-ram detekterad

Bit 0 PE: Paritetsfel

Denna bit sätts av hårdvara när ett paritetsfel uppträder hos mottagaren. Biten återställs av en läsning från USART_SR följt av en läsning från USART_DR. Programmet måste vänta på att RXNE-biten ettställts innan PE-biten återställs. Ett avbrott genereras om PEIE = 1 i USART_CR1.

0: Inget paritetsfel

1: Paritetsfel USART_DR Dataregister

offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		Register
4							R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	RDR	USART_DR
							T9	T8	T7	T6	T5	T4	T3	T2	T1	T0	TDR	

Bits 8: 0 DR [8: 0]:

Innehåller tecknet som tas emot eller sänds. USART_DR har alltså en dubbel funktion (läs och skriv) eftersom det består av två olika fysiska register, ett för sändning (TDR) och en för mottagning (RDR). Vid sändning med paritet aktiverat (PCE bit satt till 1 i register USART_CR1), är den mest signifikanta biten betydelselös eftersom den ersätts av pariteten för ordet. Vid mottagning med paritet aktiverat är det värde som avlästs i den mest signifikanta biten den mottagna pariteten.