1 misc

1.1 ASCII table

b_7	b_6	b_5			0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
BITS												
b_4	b_3	b_2	b_1		0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	|
1	1	0	1	13	CR	GS	-	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

1.2 ARM ISA

1.2.1 ARM Assembler Commands

Type	Mnemonic	Instruction
Arithmetic Register	**ADD**	Add
	ADDS	Add and set flags
	SUB	Subtract
	SUBS	Subtract and set flags
	CMP	Compare
	CMN	Compare negative
	NEG	Negate
	NEGS	Negate and set flags
Arithmetic Immediate	**ADDI**	Add Immediate
	ADDIS	Add and set flags Immediate
	SUBI	Subtract Immediate
	SUBIS	Subtract and set flags Immediate
	CMPI	Compare Immediate
	CMNI	Compare negative Immediate
Arithmetic Extended	ADD	Add Extended Register
	ADDS	Add and set flags Extended
	SUB	Subtract Extended Register
	SUBS	Subtract and set flags Extended
	CMP	Compare Extended Register
	CMN	Compare negative Extended
Arithmetic with Carry	ADC	Add with carry
	ADCS	Add with carry and set flags
	SBC	Subtract with carry
	SBCS	Subtract with carry and set flags
	NGC	Negate with carry
	NGCS	Negate with carry and set flags
Logical Register	**AND**	Bitwise AND
	ANDS	Bitwise AND and set flags
	ORR	Bitwise inclusive OR
	EOR	Bitwise exclusive OR
	BIC	Bitwise bit clear
	BICS	Bitwise bit clear and set flags
	ORN	Bitwise inclusive OR NOT
	EON	Bitwise exclusive OR NOT
	MVN	Bitwise NOT
	TST	Test bits

Type	Mnemonic	Instruction
Logical Immediate	**ANDI**	Bitwise AND Immediate
	ANDIS	Bitwise AND and set flags Immediate
	ORRI	Bitwise inclusive OR Immediate
	EORI	Bitwise exclusive OR Immediate
	TSTI	Test bits Immediate
Shift Register Shift Immed	**LSL**	Logical shift left Immediate
	LSR	Logical shift right Immediate
	ASR	Arithmetic shift right Immediate
	ROR	Rotate right Immediate
	LSRV	Logical shift right register
	LSLV	Logical shift left register
	ASRV	Arithmetic shift right register
	RORV	Rotate right register
Move Wide Immed iate	**MOVZ**	Move wide with zero
	MOVK	Move wide with keep
	MOVN	Move wide with NOT
	MOV	Move register
Bit Field Insert & Extract	BFM	Bitfield move
	SBFM	Signed bitfield move
	UBFM	Unsigned bitfield move (32-bit)
	BFI	Bitfield insert
	BFXIL	Bitfield extract and insert low
	SBFIZ	Signed bitfield insert in zero
	SBFX	Signed bitfield extract
	UBFIZ	Unsigned bitfield insert in zero
	UBFX	Unsigned bitfield extract
	EXTR	Extract register from pair
Sign Extend	*SXTB*	Sign-extend byte
	SXTH	Sign-extend halfword
	SXTW	Sign-extend word
	UXTB	Unsigned extend byte
	UXTH	Unsigned extend halfword
Bit Operation	CLS	Count leading sign bits
	CLZ	Count leading zero bits
	RBIT	Reverse bit order
	REV	Reverse bytes in register
	REV16	Reverse bytes in halfwords
	REV32	Reverses bytes in words

Type	Mnemonic	Instruction
Unscaled	**LDUR**	Load register (unscaled offset)
	LDURB	Load byte (unscaled offset)
	LDURSB	Load signed byte (unscaled offset)
	LDURH	Load halfword (unscaled offset)
	LDURSH	Load signed halfword (unscaled offset)
	LDURSW	Load signed word (unscaled offset)
	STUR	Store register (unscaled offset)
	STURB	Store byte (unscaled offset)
	STURH	Store halfword (unscaled offset)
	STURW	Store word (unscaled offset)
	LDA	Load address
Scaled, Extended, Pre-& Post-Indexed	LDR	Load register
	LDRB	Load byte
	LDRSB	Load signed byte
	LDRH	Load halfword
	LDRSH	Load signed halfword
	LDRSW	Load signed word
	STR	Store register
	STRB	Store byte
	STRH	Store halfword

Type	Mnemonic	Instruction
Exclusive	**LDXR**	Load Exclusive register
	LDXRB	Load Exclusive byte
	LDXRH	Load Exclusive halfword
	LDXP	Load Exclusive Pair
	STXR	Store Exclusive register
	STXRB	Store Exclusive byte
	STXRH	Store Exclusive halfword
	STXP	Store Exclusive Pair
Exclusive Aquire/Release	LDAXR	Load-aquire Exclusive register
	LDAXRB	Load-aquire Exclusive byte
	LDAXRH	Load-aquire Exclusive halfword
	LDAXP	Load-aquire Exclusive Pair
	STLXR	Store-release Exclusive register
	STLXRB	Store-release Exclusive byte
	STLXRH	Store-release Exclusive halfword
	STLXP	Store-release Exclusive Pair
Pair	LDP	Load Pair
	LDPSW	Load Pair signed words
	STP	Store Pair
PC	ADRP	Compute address of 4KB page at a PC-relative offset
	ADR	Compute address of label at a PC-relative offset

Type	Mnemonic	Instruction
Conditional Branch	**B.cond**	Branch conditionally
	CBNZ	Compare and branch if nonzero
	CBZ	Compare and branch if zero
	TBNZ	Test bit and branch if nonzero
	TBZ	Test bit and branch if zero
Unconditional Branch	**B**	Branch unconditionally
	BL	Branch with link
	BLR	Branch with link to register
	BR	Branch to register
	RET	Return from subroutine

Type	Mnemonic	Instruction
Conditional Select	CSEL	Conditional select
	CSINC	Conditional select increment
	CSINV	Conditional select inversion
	CSNEG	Conditional select negation
	CSET	Conditional set
	CSETM	Conditional set mask
	CINC	Conditional increment
	CINV	Conditional invert
	CNEG	Conditional negate
Conditional Compare	CCMP	Conditional compare register
	CCMPI	Conditional compare immediate
	CCMN	Conditional compare negative register
	CCMNI	Conditional compare negative immediate

Type	Mnemonic	Instruction
Non-cache	LDNP	Load Non-temporal Pair
	STNP	Store Non-temporal Pair
Barrier	CLREX	Clear exclusive monitor
	DSB	Data synchronization barrier
	DMB	Data memory barrier
	ISB	Instruction synchronization barrier
CRC	CRC32B	CRC-32 sum from byte
	CRC32H	CRC-32 sum from halfword
	CRC32W	CRC-32 sum from word
	CRC32X	CRC-32 sum from doubleword
	CRC32CB	CRC-32C sum from byte
	CRC32CH	CRC-32C sum from halfword
	CRC32CW	CRC-32C sum from word
	CRC32CX	CRC-32C sum from doubleword
Crypto	AESD	AES single round decryption
	AESE	AES single round encryption
	AESIMC	AES inverse mix columns
	AESMC	AES mix columns
	PMULL	Polynomial multiply long
	SHA1C	SHA1 hash update (choose)
	SHA1H	SHA1 fixed rotate
	SHA1M	SHA1 hash update (majority)
	SHA1P	SHA1 hash update (parity)
	SHA1SU0	SHA1 schedule update 0
	SHA1SU1	SHA1 schedule update 1
	SHA256H	SHA256 hash update (part 1)
	SHA256H2	SHA256 hash update (part 2)
	SHA256SU0	SHA256 schedule update 0
	SHA256SU1	SHA256 schedule update 1
Sys Reg	MRS	Move system register to general-purpose register
	MSR	Move general-purpose register or immediate to system register

Type	Mnemonic	Instruction
Unprivileged	LDTR	Load Unprivileged register
	LDTRB	Load Unprivileged byte
	LDTRSB	Load Unprivileged signed byte
	LDTRH	Load Unprivileged halfword
	LDTRSH	Load Unprivileged signed halfword
	LDTRSW	Load Unprivileged signed word
	STTR	Store Unprivileged register
	STTRB	Store Unprivileged byte
	STTRH	Store Unprivileged halfword
Exception	BRK	Software breakpoint instruction
	HLT	Halting software breakpoint instruction
	HVC	Generate exception targeting Exception level 2
	SMC	Generate exception targeting Exception level 3
	SVC	Generate exception targeting Exception level 1
	ERET	Exception return using current ELR and SPSR
Debug	DCPS1	Debug switch to Exception level 1
	DCPS2	Debug switch to Exception level 2
	DCPS3	Debug switch to Exception level 3
	DRPS	Debug restore PE state
System	SYS	System instruction
	SYSL	System instruction with result
	IC	Instruction cache maintenance
	DC	Data cache maintenance
	AT	Address translation
	TLBI	TLB Invalidate
Hint	NOP	No operation
	YIELD	Yield hint
	WFE	Wait for event
	WFI	Wait for interrupt
	SEV	Send event
	SEVL	Send event local
	HINT	Unallocated hint

1.2.2 Opcode

Instruction	Opcode	Opcode Size	11-bit opcode range		Instruction Format
			Start	End	
B	000101	6	160	191	B - format
STURB	00111000000	11	448		D - format
LDURB	00111000010	11	450		D - format
B.cond	01010100	8	672	679	CB - format
ORRI	1011001000	10	712	713	I - format
EORI	1101001000	10	840	841	I - format
STURH	01111000000	11	960		D - format
LDURH	01111000010	11	962		D - format
AND	10001010000	11	1104		R - format
ADD	10001011000	11	1112		R - format
ADDI	1001000100	10	1160	1161	I - format
ANDI	1001001000	10	1168	1169	I - format
BL	100101	6	1184	1215	B - format
ORR	10101010000	11	1360		R - format
ADDS	10101011000	11	1368		R - format
ADDIS	1011000100	10	1416	1417	I - format
CBZ	10110100	8	1440	1447	CB - format
CBNZ	10110101	8	1448	1455	CB - format
STURW	10111000000	11	1472		D - format
LDURSW	10111000100	11	1476		D - format
STXR	11001000000	11	1600		D - format
LDXR	11001000010	11	1602		D - format
EOR	11101010000	11	1616		R - format
SUB	11001011000	11	1624		R - format
SUBI	1101000100	10	1672	1673	I - format
MOVZ	110100101	9	1684	1687	IM - format
LSR	11010011010	11	1690		R - format
LSL	11010011011	11	1691		R - format
BR	11010110000	11	1712		R - format
ANDS	11101010000	11	1872		R - format
SUBS	11101011000	11	1880		R - format
SUBIS	1111000100	10	1928	1929	I - format
ANDIS	1111001000	10	1936	1937	I - format
MOVK	111100101	9	1940	1943	IM - format
STUR	11111000000	11	1984		D - format
LDUR	11111000010	11	1986		D - format

1.3 ARM buses

1.3.1 AHB/AMBA signals

AHB bus signals

Table 2-1 AMBA AHB signals

Name	Source	Description
HCLK Bus clock	Clock source	This clock times all bus transfers. All signal timings are related to the rising edge of **HCLK.**
HRESETn Reset	Reset controller	The bus reset signal is active LOW and is used to reset the system and the bus. This is the only active LOW signal.
HADDR[31:0] Address bus	Master	The 32-bit system address bus.
HTRANS[1:0] Transfer type	Master	Indicates the type of the current transfer, which can be NONSEQUENTIAL, SEQUENTIAL, IDLE or BUSY.
HWRITE Transfer direction	Master	When HIGH this signal indicates a write transfer and when LOW a read transfer.
HSIZE[2:0] Transfer size	Master	Indicates the size of the transfer, which is typically byte (8-bit), halfword (16-bit) or word (32-bit). The protocol allows for larger transfer sizes up to a maximum of 1024 bits.
HBURST[2:0] Burst type	Master	Indicates if the transfer forms part of a burst. Four, eight and sixteen beat bursts are supported and the burst may be either incrementing or wrapping.
HPROT[3:0] Protection control	Master	The protection control signals provide additional information about a bus access and are primarily intended for use by any module that wishes to implement some level of protection. The signals indicate if the transfer is an opcode fetch or data access, as well as if the transfer is a privileged mode access or user mode access. For bus masters with a memory management unit these signals also indicate whether the current access is cacheable or bufferable.

Read/Write bus signals

Name	Source	Description
HWDATA[31:0] Write data bus ***master to slave***	Master	The write data bus is used to transfer data from the master to the bus slaves during write operations. A minimum data bus width of 32 bits is recommended. However, this may easily be extended to allow for higher bandwidth operation.
HSELx Slave select ***„chip“ select***	Decoder	Each AHB slave has its own slave select signal and this signal indicates that the current transfer is intended for the selected slave. This signal is simply a combinatorial decode of the address bus.
HRDATA[31:0] Read data bus ***slave to master***	Slave	The read data bus is used to transfer data from bus slaves to the bus master during read operations. A minimum data bus width of 32 bits is recommended. However, this may easily be extended to allow for higher bandwidth operation.
HREADY Transfer done ***NOT wait***	Slave	When HIGH the HREADY signal indicates that a transfer has finished on the bus. This signal may be driven LOW to extend a transfer. Note: Slaves on the bus require HREADY as both an input and an output signal.
HRESP[1:0] Transfer response	Slave	The transfer response provides additional information on the status of a transfer. Four different responses are provided, OKAY, ERROR, RETRY and SPLIT.

Arbitration signals

Table 2-2 Arbitration signals

Name	Source	Description
HBUSREQx Bus request	Master	A signal from bus master x to the bus arbiter which indicates that the bus master requires the bus. There is an **HBUSREQx** signal for each bus master in the system, up to a maximum of 16 bus masters.
HLOCKx Locked transfers	Master	When HIGH this signal indicates that the master requires locked access to the bus and no other master should be granted the bus until this signal is LOW.
HGRANTx Bus grant	Arbiter	This signal indicates that bus master x is currently the highest priority master. Ownership of the address/control signals changes at the end of a transfer when **HREADY** is HIGH, so a master gets access to the bus when both **HREADY** and **HGRANTx** are HIGH.
HMASTER[3:0] Master number	Arbiter	These signals from the arbiter indicate which bus master is currently performing a transfer and is used by the slaves which support SPLIT transfers to determine which master is attempting an access. The timing of **HMASTER** is aligned with the timing of the address and control signals.
HMASTLOCK Locked sequence	Arbiter	Indicates that the current master is performing a locked sequence of transfers. This signal has the same timing as the **HMASTER** signal.
HSPLITx[15:0] Split completion request	Slave (SPLIT-capable)	This 16-bit split bus is used by a slave to indicate to the arbiter which bus masters should be allowed to re-attempt a split transaction. Each bit of this split bus corresponds to a single bus master.

APB signals

Table 2-4 AMBA APB signals

Name	Description
PCLK Bus clock	The rising edge of **PCLK** is used to time all transfers on the APB.
PRESETn APB reset	The APB bus reset signal is active LOW and this signal will normally be connected directly to the system bus reset signal.
PADDR[31:0] APB address bus	This is the APB address bus, which may be up to 32-bits wide and is driven by the peripheral bus bridge unit.
PSELx APB select	A signal from the secondary decoder, within the peripheral bus bridge unit, to each peripheral bus slave x. This signal indicates that the slave device is selected and a data transfer is required. There is a **PSELx** signal for each bus slave.
PENABLE APB strobe	This strobe signal is used to time all accesses on the peripheral bus. The enable signal is used to indicate the second cycle of an APB transfer. The rising edge of **PENABLE** occurs in the middle of the APB transfer.
PWRITE APB transfer direction	When HIGH this signal indicates an APB write access and when LOW a read access.
PRDATA APB read data bus	The read data bus is driven by the selected slave during read cycles (when **PWRITE** is LOW). The read data bus can be up to 32-bits wide.
PWDATA APB write data bus	The write data bus is driven by the peripheral bus bridge unit during write cycles (when **PWRITE** is HIGH). The write data bus can be up to 32-bits wide.

AHB burst signals

HTRANS[1:0]	Type	Description
00	IDLE	Indicates that no data transfer is required. The IDLE transfer type is used when a bus master is granted the bus, but does not wish to perform a data transfer. Slaves must always provide a zero wait state OKAY response to IDLE transfers and the transfer should be ignored by the slave.
01	BUSY	The BUSY transfer type allows bus masters to insert IDLE cycles in the middle of bursts of transfers. This transfer type indicates that the bus master is continuing with a burst of transfers, but the next transfer cannot take place immediately. When a master uses the BUSY transfer type the address and control signals must reflect the next transfer in the burst. The transfer should be ignored by the slave. Slaves must always provide a zero wait state OKAY response, in the same way that they respond to IDLE transfers.
10	NONSEQ	Indicates the first transfer of a burst or a single transfer. The address and control signals are unrelated to the previous transfer. Single transfers on the bus are treated as bursts of one and therefore the transfer type is NONSEQUENTIAL.
11	SEQ	The remaining transfers in a burst are SEQUENTIAL and the address is related to the previous transfer. The control information is identical to the previous transfer. The address is equal to the address of the previous transfer plus the size (in bytes). In the case of a wrapping burst the address of the transfer wraps at the address boundary equal to the size (in bytes) multiplied by the number of beats in the transfer (either 4, 8 or 16).

size and beats of burst

HSIZE[2:0] Transfer size	Master	Indicates the size of the transfer, which is typically byte (8-bit), halfword (16-bit) or word (32-bit). The protocol allows for larger transfer sizes up to a maximum of 1024 bits.

HBURST[2:0]	Type	Description
000	SINGLE	Single transfer
001	INCR	Incrementing burst of unspecified length
010	WRAP4	4-beat wrapping burst
011	INCR4	4-beat incrementing burst
100	WRAP8	8-beat wrapping burst
101	INCR8	8-beat incrementing burst
110	WRAP16	16-beat wrapping burst
111	INCR16	16-beat incrementing burst

Slave response signals

HRESP[1]	HRESP[0]	Response	Description
0	0	OKAY	When **HREADY** is HIGH this shows the transfer has completed successfully. The OKAY response is also used for any additional cycles that are inserted, with **HREADY** LOW, prior to giving one of the three other responses.
0	1	ERROR	This response shows an error has occurred. The error condition should be signalled to the bus master so that it is aware the transfer has been unsuccessful. A two-cycle response is required for an error condition.
1	0	RETRY	The RETRY response shows the transfer has not yet completed, so the bus master should retry the transfer. The master should continue to retry the transfer until it completes. A two-cycle RETRY response is required.
1	1	SPLIT	The transfer has not yet completed successfully. The bus master must retry the transfer when it is next granted access to the bus. The slave will request access to the bus on behalf of the master when the transfer can complete. A two-cycle SPLIT response is required.

1.3.2 AXI

Read/write adress-channel signals[1]

Signal description	Write addr. ch.	Read addr. ch.
Address ID	AWID	ARID
Address of the first beat of the burst	AWADDR	ARADDR
Number of beats inside the burst	AWLEN	ARLEN
Size of each beat	AWSIZE	ARSIZE
Type of the burst	AWBURST	ARBURST
Lock type, to provide atomic operations	AWLOCK	ARLOCK
Memory type	AWCACHE	ARCACHE
Protection type	AWPROT	ARPROT
Quality of Service of the transaction	AWQOS	ARQOS

[1]Descriptions shorted so table can fit on screen

Region identifier	AWREGION	ARREGION
User-defined data	AWUSER	ARUSER
xVALID handshake signal	AWVALID	ARVALID
xREADY handshake signal	AWREADY	ARREADY

Read/Write data-channel signals

Signal description	Write dat. ch.	Read dat. ch.
Data ID, to identify multiple streams over a single channel	WID	RID
Read/Write data	WDATA	RDATA
Read response, status of the current RDATA signal	-	RRESP
Byte strobe, which bytes of WDATA are valid	WSTRB	-
Last beat identifier	WLAST	RLAST
User-defined data	WUSER	RUSER
xVALID handshake signal	WVALID	RVALID
xREADY handshake signal	WREADY	RREADY

Write response signals

Signal description	Write resp. ch.
Write response ID	BID
Write response, to specify the status of the burst	BRESP
User-defined data	BUSER
xVALID handshake signal	BVALID
xREADY handshake signal	BREADY

Slave response channel (uses write repsonse channel)

OKAY	The answer of the slave if a normal access has been successful.
EXOKAY	The answer of the slave if a normal exclusive access has been successful. An exclusive access is an access where only one master is allowed to access this slave. It is not required, that the bus is occupied the whole time.
SLVERR	This error will be set, when the transaction has been faulty.
DECERR	This error will be signalled by the interconnect to the master if it couldn't find the slave under the given adress. occupied the whole time.

1.3.3 Burst

ARBURST[1:0], AWBURST[1:0]	Burst Type
00_2	FIXED
01_2	INCR
10_2	WRAP
11_2	Reserved

ARSIZE[2:0], AWSIZE[2:0]	Bytes in transfer
3 bits	1 - 128 ($2^{ARSIZE/AWSIZE}$)

ARLEN[3:0], AWLEN[3:0]	Number of data transfers
4 bits	1 - 16 (2^{1-4})