ARM hardware

reference manual

ARM Evaluation System

Acorn OEM Products

ARM hardware

Part No 0448,006
Issue No 1.0
4 August 1986

Within this publication the term 'BBC' is used as an abbreviation for 'British Broadcasting Corporation'.

- The manual is provided on an 'as is' basis except for warranties described in the software licence agreement if provided.
- The software and this manual are protected by Trade secret and Copyright laws.

The product described in this manual is subject to continuous developments and improvements. All particulars of the product and its use (including the information in this manual) are given by Acorn Computers in good faith.

There are no warranties implied or expressed including but not limited to implied warranties or merchantability or fitness for purpose and all such warranties are expressly and specifically disclaimed.

In case of difficulty please contact your supplier. Every step is taken to ensure that the quality of software and documentation is as high as possible. However, it should be noted that software cannot be written to be completely free of errors. To help Acorn rectify future versions, suspected deficiencies in software and documentation, unless notified otherwise, should be notified in writing to the following address:

Customer Services Department,
Acorn Computers Limited,
645 Newmarket Road,
Cambridge
CB5 8PD

ISBN 1 85250 002

Published by:
Acorn Computers Limited, Fulbourn Road, Cherry Hinton, Cambridge CB1 4JN, UK

Important Information

Wiring the Mains Plug

WARNING: The ARM Evaluation System must be earthed.

The wires in the mains lead are coloured in accordance with the following code:

Green and yellow	Earth
Blue	Neutral
Brown	Live

As the colours of the wires may not correspond with the coloured markings identifying the terminals in your plug, proceed as follows:

- The wire which is coloured *green and yellow* must be connected to the terminal in your plug which is marked by the letter E, or by the safety earth symbol or which is identified by being coloured green, or green and yellow.
- The wire which is coloured *blue* must be connected to the terminal which is marked with the letter N, or coloured black.
- The wire which is coloured *brown* must be connected to the terminal which is marked with the letter L, or coloured red.

If the socket outlet available is not suitable for the plug supplied, the plug should be cut off and the appropriate one fitted and wired as previously noted. The moulded plug which was cut off should be disposed of as it would be a potential shock hazard if it were to be plugged in with the cut off end of the mains cord exposed. The moulded plug must be used with the fuse and fuse carrier firmly in place. The fuse carrier is of the same basic colour (though not necessarily the same shade of that colour) as the coloured insert in the base of the plug. Different manufacturers' plugs and fuse carriers are not interchangeable. In the event of loss of the fuse carrier, the moulded plug MUST NOT be used. Either replace the moulded plug with another conventional plug (wired as previously described) or obtain a replacement fuse carrier from an authorised Acorn dealer. In the event of the fuse blowing, it should be replaced, after clearing any faults, with a 5 amp fuse that is ASTA approved to BSI 1362.

Exposure

Like all electronic equipment, the ARM Evaluation System should not be exposed to direct sunlight or moisture for long periods.

Contents

1. Introduction 1
   1.1 Features 2
   1.2 Performance 2
2. Standard specifications 4
   2.1 Physical description 4
   2.2 Power requirements 4
   2.3 Temperatures 4
3. Functional description 5
   3.1 ARM block diagram 5
   3.2 Pin connections, pins 1 to 42 6
   3.3 ARM element functions 8
   3.4 ARM operational description 10
4. Signal description 14
   4.1 D.C. characteristics 14
   4.2 Absolute maximum ratings 15
   4.3 Signal definitions 15
   4.4 Timing information, signal dwell times 20
   4.5 Timing waveforms 22
5. The ARM co-processor 23
   5.1 Description of the ICs 23
      5.1.1 IC 1 PAL20L8 24
      5.1.2 IC 2 PAL20L8 27
      5.1.3 IC 3 PAL20L8 30
      5.1.4 IC 9 PAL20R8 31
      5.1.5 IC 18 PAL20R8 34
   5.2 Timing diagrams 36
   5.3 ARM co-processor circuit diagram 37
6. Appendix A 38
   6.1 ARM instruction set 38
7. Appendix B 40
   7.1 Document references 40

1. Introduction

The ARM (Acorn RISC Machine) is a general purpose 32-bit single-chip microprocessor which uses a Reduced Instruction Set Computer architecture in order to achieve high performance.

The Arm contains a 32-bit data bus, 26-bit address bus and a bank of 25 registers, each 32 bits wide.

The instruction set, comprising five basic instruction types, each with an associated 4-bit condition code, is hard wired.

Pipelining is employed, so that all parts of the processing and memory system can be used during every cycle, when executing consecutive register-to-register instructions.

The technology used is 3-micrometres double-level metal CMOS; the chip size is 50 square millimetres, packaged in an 84-pin leadless carrier.

1.1 Features

- 32-bit architecture
- 32-bit data bus
- 26-bit address bus
- 64-MByte uniform address space
- Simple but powerful instruction set
- Good high-level language compiler support
- Support for virtual memory systems
- Fast interrupt response for real-time applications (average interrupt latency less than 2µS, worst case less than 6µS)
- Average execution rate 3 million instructions per second (MIPS)
- Low power consumption (0.1 W typical)
- Single +5 V supply
- 84-pin JEDEC B leadless chip carrier

1.2 Performance

The ARM microprocessor has been specifically designed for high-performance functions such as real-time artificial intelligence and high-level language applications. The Acorn chip is smaller and the architecture simpler than conventional microprocessors, yet its execution rate of 3 MIPS is one of the fastest available.

The ARM supports virtual memory, has a small optimised instruction set hard wired into a programmable logic array, a heavily pipelined processor, dedicated registers to handle interrupts and a high memory-to-processor bandwidth.

The instructions are all 32 bits wide (one word) and the instruction set consists of five basic types:

- branch (and branch with link)
- data processing
- single data transfer
- block data transfer
- supervisor calls.

The ARM utilises pipelining techniques to gain greater efficiency in the manipulation of instructions. During each processor cycle, one instruction can control the data path while the system decodes a second instruction for the following cycle and fetches a third from memory.

Another performance advantage is the processor's ability to support memory operation in burst (or page) mode. In burst mode, data can be continuously streamed to or from memory, at least twice as fast as in random access mode, depending upon the effect of memory-to-processor interaction.

The ARM performance may be summarised as being:

- approximately 3 MIPS average, using 150 nanoseconds row access DRAMS (evaluation system measured results)
- 8 MIPS peak (first prototype).

This is equivalent to:

- 2 to 4 times DEC VAX 11/780 running high-level benchmarks
- 10 times IBM PC AT running BASIC benchmarks
- A 16.67 MHz Motorola 68020.

(This performance was measured on an ARM Evaluation System.)

The average interrupt latency is less than 2µS; the maximum latency is less than 6µS.

2. Standard specifications

2.1 Physical description

The ARM is currently available in an 84-pin JEDEC B ceramic carrier.

A suitable socket, such as AMP, part number 55225-1, may be used for mounting the carrier onto a printed circuit board.

2.2 Power requirements

	absolute maximum rating	nominal
Vcc (supply voltage with respect to Vss)	+6.0 V d.c.*	+5.0 V d.c.
Power dissipation	1 W	

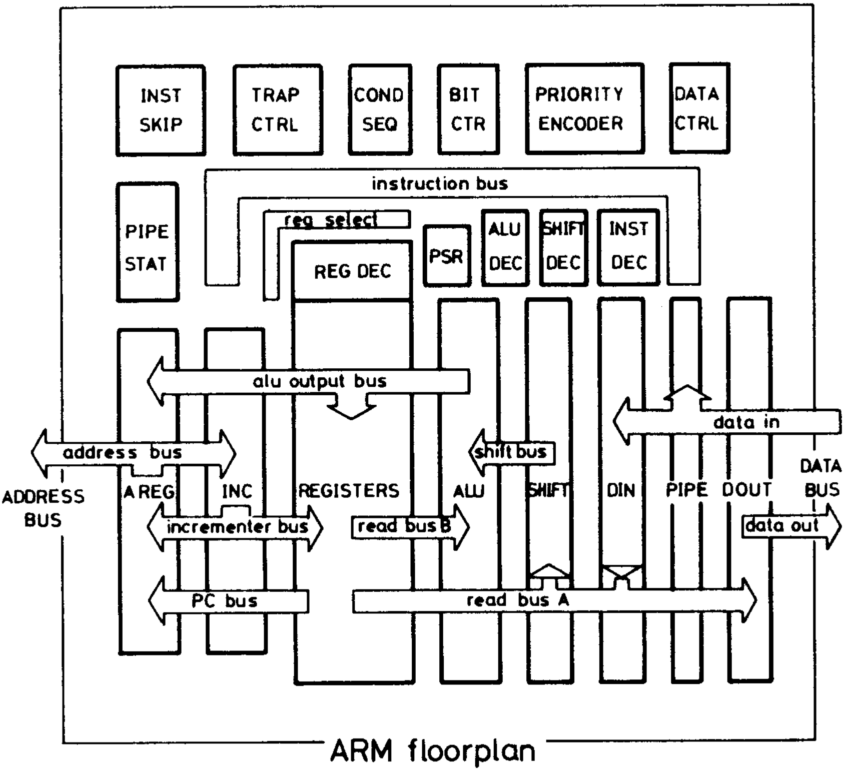
* Note: absolute maximum ratings indicate limits beyond which permanent damage may occur. Operation at these limits is not guaranteed and should be limited to those conditions specified in section 4.1, D.C. characteristics.

2.3 Temperatures

–40 to +70 °C storage without damage.

3. Functional description

3.1 ARM block diagram



3.2 Pin connections, pins 1 to 42

function	name	input/ output	pin number
processor clock	ph2	I	1
processor clock	ph1	I	2
write/not read	rw	O	3
not opcode fetch	opc	O	4
not next cycle memory req.	mreq	O	5
memory abort	abort	I	6
not interrupt request	irq	I	7
not fast interrupt request	fiq	I	8
reset	reset	I	9
not memory translate	trans	O	10
supply voltage	vcc1	I	11
ground	vss1	I	12
not processor mode bit 1	m1	O	13
not processor mode bit 0	m0	O	14
next cycle sequence indicator	seq	O	15
address latch enable	ale	I	16
address line 25	a25	T	17
address line 24	a24	T	18
address line 23	a23	T	19
address line 22	a22	T	20
address line 21	a21	T	21
address line 20	a20	T	22
address line 19	a19	T	23
address line 18	a18	T	24
address line 17	a17	T	25
address line 16	a16	T	26
address line 15	a15	T	27
address line 14	a14	T	28
address line 13	a13	T	29
address line 12	a12	T	30
address line 11	a11	T	31
supply voltage	vcc2	I	32
ground	vss2	I	33

function	**name**	**input/ output**	**pin number**
address line 10	a10	T	34
address line 9	a9	T	35
address line 8	a8	T	36
address line 7	a7	T	37
address line 6	a6	T	38
address line 5	a5	T	39
address line 4	a4	T	40
address line 3	a3	T	41
address line 2	a2	T	42
address line 1	a1	T	43
address line 0	a0	T	44
address bus Tri-state enable	abe	I	45
data line 0	d0	B	46
data line 1	d1	B	47
data line 2	d2	B	48
data line 3	d3	B	49
data line 4	d4	B	50
data line 5	d5	B	51
data line 6	d6	B	52
data line 7	d7	B	53
ground	vss3	I	54
supply voltage	vcc3	I	55
data line 8	d8	B	56
data line 9	d9	B	57
data line 10	d10	B	58
data line 11	d11	B	59
data line 12	d12	B	60
data line 13	d13	B	61
data line 14	d14	B	62
data line 15	d15	B	63
data line 16	d16	B	64
data line 17	d17	B	65
data line 18	d18	B	66
data line 19	d19	B	67
data line 20	d20	B	68

function	name	input/ output	pin number
data line 21	d21	B	69
data line 22	d22	B	70
data line 23	d23	B	71
data line 24	d24	B	72
data line 25	d25	B	73
data line 26	d26	B	74
ground	vss4	I	75
supply voltage	vcc4	I	76
data line 27	d27	B	77
data line 28	d28	B	78
data line 29	d29	B	79
data line 30	d30	B	80
data line 31	d31	B	81
not connected	-	-	82
data bus enable	dbe	I	83
word/not byte transfer	bw	O	84

symbols:

I = input to chip
O = output from chip
B = bi-directional
T = tri-state output.

3.3 ARM element functions

Referring to the block diagram in section 3.1, the functions of the main units are as follows:

element	function
alu	arithmetic logic unit
registers	a bank of 25 32-bit registers
shift	a 32-bit barrel shifter to assist arithmetic and logical register operations with a rotate capability

element	**function**
data in/out	32-bit data bus
din	the data input control logic, which extracts the required field from the incoming data or instruction
dout	the data output control block, which replicates a byte across the data bus for byte write operations
addbus	26-bit address bus
inc	the address incrementer
areg	the current address register
pipe	the instruction pipeline
inst skip	controls the skipping (non-execution) of instructions which do not meet the required condition codes
trap cntrl	handles the synchronisation and prioritisation of interrupts, exceptions, aborts and reset
cond seq	evaluates the instruction condition field and controls the instruction cycle sequence
bit ctr	counts the number of 1s in the 16-bit field used by load and store multiple instructions
priority encoder	finds the least significant 1 in the 16-bit field used by load and store multiple operations
data ctrl	controls the flow of data into the processor
pipe stat	keeps the status of instructions in the pipeline, differentiating between valid instructions and those which cause an abort

element	**function**
instruction bus	carries the current instruction for decoding by the various control blocks
reg select	selects the fields in the current instruction which define the registers to be used
reg dec	decodes the selected fields to access particular registers
psr	the program status register, which contains the ALU flags, the interrupt masks and the processor mode bits
alu dec	a PLA (programmable logic array) which controls the function of the ALU
shift dec	the barrel shifter control
inst dec	a PLA which performs the top-level instruction decoding.

3.4 ARM operational description

The ARM is a 32-bit single chip microprocessor based on a reduced instruction set architecture.

The chip runs on a non-overlapping two-phase clock and all data path operations take place in one clock cycle (which is clock phase 1 plus clock phase 2).

The heart of the chip is the register bank, which contains 25 32-bit registers of which 16 are visible to the programmer. The remainder of these registers are used to support the supervisor, the interrupt and the fast interrupt modes. The register bank contains two read buses and one write bus. The two read buses enable both ALU operands to be fetched from the register bank simultaneously:

- one operand is passed through the barrel shifter before going into the ALU (the fetches taking place during clock phase 1). The result may be written back to a register during clock phase 2
- the second (shifted) operand may be obtained from an immediate field in the instruction, rather than from a register.

The memory address is held in the address register and, associated with this register, there is a dedicated address incrementer. The address for the next memory cycle may come from the ALU or be forced to an exception value, but, under normal operation, it is taken from the incrementer. When the incrementer is the address source, this fact is indicated by asserting the SEQ pin. External memory control can then predict the next address and take decisions as to whether address translation is necessary or DRAM page mode can be used.

Note: using DRAM page mode enables the cycle to proceed at double speed (or greater) and saves power compared to a full RAM access. When executing typical program code, page mode accesses are used for 70% to 90% of all memory cycles.

The instruction pipeline holds instructions awaiting execution. It is synchronous, fetches instructions at defined times, and is of minimum length to keep all sections of the processor busy during consecutive register to register instructions. As one such instruction is being executed on the data path, the next is being decoded while a third is being fetched from memory. Each instruction occupies a part of the processor for three cycles, but the pipelining technique used allows the execution of one instruction per cycle.

Output word data is sent as 32-bit words, aligned on the data bus. Byte data is replicated four times across the data bus and the correct memory byte can be written by activating only the relevant column address strobe (CAS). The bw (word/not byte) signal indicates a byte transfer and a0 & a1 (address lines) indicate the byte within the word.

Input word-aligned data is transferred to the target register. Input bytes are field extracted, zero extended in the data input block (din) and rotated into the lowest byte position by the barrel shifter, before being placed in the target register.

The priority encoder is used only in block transfer instructions. These allow any defined register subset to be transferred into successive memory words during consecutive cycles using sequential memory modes. Register saving on subroutine entry therefore exploits the most efficient memory transfer mode and this can include stacking the return address from the link register. Restoring registers and returning may be performed by load multiple, loading the return address directly into the processor rather than via the link register.

the ARM supports eight exceptions, five caused by external hardware, three by internal hardware

External hardware exceptions:

(1) the asynchronous reset signal clears the current instruction and forces execution from location 0

(2) interrupt (irq) is synchronised and, when enabled, forces the processor to begin execution at a fixed memory location on completion of the current instruction

(3) fast interrupt (fiq), as described in (2) above, irq

(4) two (abort) exceptions which modify the consequences of the current instruction to ensure that a restart will be possible before forcing the execution address the forced address depends on whether the abort was the result of a data transfer or instruction fetch.

Internal hardware exceptions:

(1) supervisor calls are forced to a fixed memory location and enter supervisor mode. This is a protected state and may only be entered from user mode via an exception, allowing trusted software to take control in a system with protected memory

(2) Undefined instructions are identical to supervisor calls except that they use a different exception location. They are (by convention) reserved for future expansion and the trap may be used for the emulation of future additions to the instruction set

(3) the address exception trap, caused by attempts to access data outside the 64-Mbyte addressable range.

The interrupt priorities are:

reset (highest)
address exception
data abort
fiq
irq
prefetch abort
undefined instruction
software interrupt (lowest)

Note: not all exceptions can occur at once. Address exception and Data abort are mutually exclusive, as are Undefined instruction and Software interrupt. A summary list and a description of the instruction set is given in appendix A.

4. Signal description

4.1 D.C. characteristics

name	min	typical	max	unit	conditions
Vil	-0.30		+0.80	V	except clock
Vih	+2.40		Vcc +0.30	V	except clock
Vcil	-0.30		+0.30	V	clock
Vcih	Vcc -0.30		Vcc +0.30	V	clock
Vol			+0.50	V	Iol = -3.60 mA
Voh	Vcc -0.50			V	Ioh = +3.00 mA
Vcc	+4.75	+5.00	+5.25	V	
Icc		20.00		mA	measured at Vcc = 5.00 V

symbols

Vil	signal input low voltage
Vih	signal input high voltage
Vcil	clock input low voltage
Vcih	clock input high voltage
Vol	signal output low voltage
Voh	signal output high voltage
Iol	signal output low current
Ioh	signal output high current
Vcc	supply voltage
Vss	ground reference
Icc	supply current

4.2 Absolute maximum ratings

Power supply to ground reference (Vss to Vcc)	-0.5 to +6.0 V
Temperature under bias	0 °C to 70 °C
Storage temperature	-40 °C to +125 °C
Input or output levels	Vss-0.5 V to Vcc+0.5 V
Power dissipation	1 W

Note: absolute maximum ratings indicate limits beyond which permanent damage may occur. Operation at these limits is not guaranteed and should be limited to those conditions specified in the d.c. characteristics table given above.

4.3 Signal definitions

The input clock lines (ph1, ph2)

The phase relationship of these lines is shown in the timing diagram on page 22 and they should swing cleanly between Vss and Vdd, with no overshoot, as they are taken directly on to the ARM without input buffering. They may be driven by 74 HC (or similar) series circuits and should not overlap at their +2.0 V points. A processor cycle is defined as <phase 1> followed by <phase 2>.

<phase 1> is the period during which line ph1 is high.

<phase 2> is the period during which line ph2 is high.

Write/not read (rw)

This signal, when high, indicates a processor write cycle; when low, a read cycle. This signal becomes valid during phase 2 of the cycle before that to which it refers, remaining valid to the end of phase 1 of the referenced cycle.

Operation code (opc)

This signal, when low, indicates that the processor is fetching an instruction. The signal becomes valid during phase 2 of the previous cycle, remaining valid through phase 1 of the referenced cycle.

Memory request (mreq)

This signal, when low, indicates that the processor requires memory access during the following cycle. The signal becomes valid during phase 1, remaining valid through phase 2 of the cycle preceding that to which it refers.

Abort (abort)

This is an input which allows the memory system to tell the processor that a requested access is not allowed. The signal must be valid before the end of phase 1 of the cycle during which the memory transfer is attempted.

Interrupt request (irq)

This is an asynchronous interrupt request to the processor which causes it to be interrupted if taken low when the appropriate enable in the processor is active. The signal is level sensitive and must be held low until a suitable response is received from the processor.

Fast interrupt request (fiq)

As irq, but with higher priority. May be taken low asynchronously to interrupt the processor when the appropriate enable is active.

Reset (reset)

This is a level sensitive input signal which is used to start the processor from a known address. A high level will cause the instruction being executed to terminate abnormally. When the reset signal becomes low for at least one clock cycle, the processor will re-start from address 0. During the period when reset is held high, the processor will perform dummy instruction fetches with the address incrementing from the point where reset was activated. The address value will overflow to zero if reset is held beyond the maximum address limit.

Translate (trans)

When this signal is low it indicates that the processor is in user mode, or that the supervisor is using a single transfer instruction with the force translate bit active. It may be used to tell memory management hardware when translation of the addresses should be turned on.

Processor mode bits (m1,m0)

These are output signals which are the inverses of the internal status bits indicating the processor operation mode.

m1	m0	mode
0	0	supervisor
0	1	irq
1	0	fiq
1	1	user

Note: this table gives the values at the pins; the internal bits are inverted with respect to this table.

Sequential (seq)

This is an output signal. It will become high when:

- the address for the next cycle is being generated in the address incrementer, so will be equal to the present address (in bytes) plus 4
- during a cycle which did not use memory (mreq inactive) when the address in the next cycle will be the same as the current address.

The signal becomes valid during phase 1 and remains so through phase 2 of the cycle before the cycle whose address it anticipates. It may be used, in combination with the low-order address lines, to indicate that the next cycle can use a fast memory mode (for example DRAM page mode) and/or to bypass the address translation system.

Address line enable (ale)

This input to the processor may be used to control transparent latches on the address outputs. Normally the addresses change, during phase 2, to the value required during the next cycle, but for direct interfacing to ROMs they are required to be stable to the end of phase 2. Taking ale low until the end of phase 2 will ensure that this happens. If the system does not require address lines to be held in this way, ale may be held permanently high.

Address lines (a0 to a25)

These are output processor address lines. If ale (address line enable) is high, the addresses become valid during phase 2 of the cycle before the one to which they refer and remain so during phase 1 of the referenced cycle. Their stable period may be controlled by ale, as described above.

Address bus enable (abe)

This is an input signal. When low, it puts the address bus drives into a high impedance state.

Data lines (d0 to d31)

These are bi-directional signal paths which are used for data transfers between the processor and external memory, as follows:

- during read cycles (when rw = 0), the input data must be valid before the end of phase 2 of the transfer cycle
- during write cycles (when rw = 1), the output data will become valid during phase 1 and remain so throughout phase 2 of the transfer cycle.

Data bus enable (dbe)

This is an input signal. When low, puts the data bus drivers into a high impedance state.

Word/not byte (bw)

This is an output signal used by the processor to indicate to the external memory system when a data transfer of a byte length is required. The signal is high for word transfers and low for byte transfers and is valid for both read and write. The signal will become valid during phase 2 of the cycle before the one during which the transfer will take place. It will remain stable throughout phase 1 of the transfer cycle. (Slow peripherals and DMA are handled by cycle stretching of the clock.)

4.4 Timing information, signal dwell times

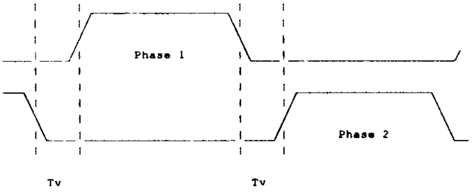
signal name	measured period (nsec) min	 max	comment
Tv	0		measured at 2 V level - no clock overlap
Tckl	70		phase 2 clock low period
Tckh	55		phase 2 clock high period
Tckl+Tckh		10	(estimate)
Taout		0	address out delay time
Tah	10		address hold time
Trws	0		read/write set up
Trwh	10		read/write hold time
Tdout		50	data out delay time
Tdoh	10		data out hold time
Tdis	5		data in set time
Tdih	10		data in hold time
Tdbe		30	data bus enable to data bus valid
Tdbz		35	data bus enable to data bus high impedance
Tabts	35		abort set-up time
Tabth	0		abort hold time
Tseqd		55	sequential indicated delay
Tseqh	10		sequential indicated hold

Note: this timing information is incomplete and is presented here for guidance only. More detailed timing information will be included when the component has been fully characterised. The figures given above indicate measured conditions required for the correct operation of the first sample devices.

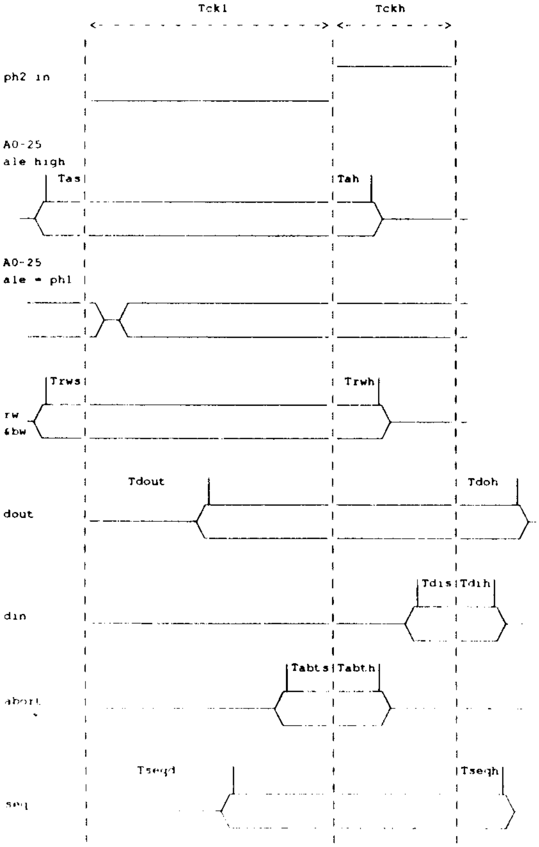
In the signal names used above, T is the time duration and the other symbols are mnemonics of the signal descriptions.

4.5 Timing waveforms

Clock phases



Memory cycle



5. The ARM co-processor

The ARM co-processor is a evaluation device intended for use with BBC Model B or BBC Master Series microcomputers.

The 6502 based processor, on the BBC Model B or BBC Master Series microcomputers, connects to the ARM processor through a high-speed data interface called the TUBE

The BBC Model B or BBC Master Series microcomputers handle all input/output operations, display memory and filing system tasks, leaving the co-processor free to handle application programs.

The co-processor is a compact, four-layer printed circuit board carrying an 84-pin JEDEC type B package, 2 MBytes of DRAM, a bootstrap ROM and an additional 2 MBytes of DRAM on a daughter board.

5.1 Description of the ICs

Referring to the circuit diagram:

IC3

The system clock generator. The crystal oscillator master clock is divided by three to produce four overlapping clock pulses.

IC1, IC2

These ICs produce all the critically timed signals on the board, by merging the outputs from IC3.

IC8

This IC is a long counter chain which times RAM refresh cycles and produces the required row addresses.

IC9

A state machine which controls the enables for the timed signals produced by ICs 1 and 2.

IC10

This is a transparent latch which delays some of the enables from IC9 to guarantee hold times in ICs 1 and 2.

IC13

This IC is the ARM processor.

IC17

This is the TUBE chip, a proprietary interface to the Acorn BBC Model B or BBC Master Series microcomputers.

IC18

This is a state machine which controls the ROM accesses.

IC19

This IC is the ROM.

IC20 to IC22

These ICs are edge triggered latches which expand data from the ROM into a 32-bit word under the control of IC18.

IC23 to IC150

These are the ICs that make up the 4 MBytes of RAM.

IC1, IC2, IC3, IC9, IC18

These are Programmable Array Logic (PAL). Descriptions of these logic elements follow, with PAL generation details and timing diagrams.

5.1.1 IC 1 PAL20L8

Timing generator 1 (A2M2PT1M1)

This PAL generates some of the system critical timing signals by merging phases of the four-phase overlapping clocks from A2M2PCKGEN, when the appropriate enables are active.

M8REF

The system reference clock.

ALEB

The processor address latch enable. The processor address latch is used to guarantee address hold time during ROM and TUBE cycles; RAM cycles use early addresses where the address latch is held open. Holds addresses to end of phase 2 except for RAM accesses

PHI1, PHI2

The two-phase, non-overlapping processor clocks. External HC04 buffers are used to ensure that these clocks have the correct voltage swing.

CAS0...CAS3

The RAM column address strobes. Individual RAM bytes are accessed by enabling only the appropriate CAS line.

```
M8REF =
  Q3                                    synthesize 8 MHz square-
  +Q2                                   wave

ALEB =
  CASENBL*/PHIENBL*Q2                   hold addresses to end of
  +CASENBL*/PHIENBL*Q1                  phase 2 except for RAM
  +CASENBL*PHWT                         accesses

PHI1 =
  /PHWT*PHIENBL*/PHI2                   processor clocks have non-
  +/PHWT*/Q2*/Q1*/PHI2                  overlap coupling

PHI2 =
  PHWT*/PHI1                            stretch phase 2 high
  +/PHIENBL*Q2*/PHI1                    ) time phase 2 edges
  +/PHIENBL*Q1*/PHI1                    )
```

```
CAS0 =
 BWL*/CASENBL*Q1*/WRL              start read early - word
 access +BWL*/CASENBL*Q2
 +BWL*/CASENBL*Q3*WRL              end read late - word access
 +/A0L*/A1L*/CASENBL*Q1*/WRL       )
 +/A0L*/A1L*/CASENBL*Q2            ) byte 0 access
 +/A0L*/A1L*/CASENBL*Q3*WRL        )
CAS1 =
 BWL*/CASENBL*Q1*/WRL              word access
 +BWL*/CASENBL*Q2                  word access
 +BWL*/CASENBL*Q3*WRL              word access
 +A0L*/A1L*/CASENBL*Q1*/WRL        )
 +A0L*/A1L*/CASENBL*Q2             ) byte 1 access
 +A0L*/A1L*/CASENBL*Q3*WRL         )
CAS2 =
 BWL*/CASENBL*Q1*/WRL              word access
 +BWL*/CASENBL*Q2                  word access
 +BWL*/CASENBL*Q3*WRL              word access
 +/A0L*A1L*/CASENBL*Q1*/WRL        )
 +/A0L*A1L*/CASENBL*Q2             ) byte 2 access
 +/A0L*A1L*/CASENBL*Q3*WRL         )
CAS3 =
 BWL*/CASENBL*Q1*/WRL              word access
 +BWL*/CASENBL*Q2                  word access
 +BWL*/CASENBL*Q3*WRL              word access
 +A0L*A1L*/CASENBL*Q1*/WRL         )
 +A0L*A1L*/CASENBL*Q2              ) byte 3 access
 +A0L*A1L*/CASENBL*Q3*WRL          )
```

5.1.2 IC 2 PAL20L8

Timing generator 2 (A2M2TIM2)

Generates the remainder (in conjunction with timing generator 1) of the system critical timing signals.

RO

The RAM row address buffer enable.

CO

The RAM column address buffer enable

RE

The RAM refresh address buffer enable.

RAS0..RAS3

The RAM row address strobes. There are four banks of RAMs and only the address bank is strobed to reduce power. The exception to this rule is during refresh, when all banks are strobed. In this case the strobes are staggered to spread the current transient.

```
RO =
 RASENB*/Q2*/Q3*RFDBL*RFDB              row address enable
 +RASENBL*Q1*RFDBL*RFDB
 +RASENBL*Q2*RFDBL*RFDB
 +RASENBL*RASENB*RFDBL*RFDB


CO =
 /RASENB*/Q2*/Q3*RFDBL                  column address enable
 +/RASENB*/RASENBL*RFDBL                (C0 comes after R0)
 +/RASENBL*Q1*RFDBL
 +/RASENBL*Q2*RFDBL


RE =
 /RFDB*/Q2*/Q3                          refresh address enable
 +/RFDB*/RFDBL
 +/RFDBL*/Q4


RAS0 =
 /RASENB*/Q2*/A20*/A21*RFDBL            memory bank 0 RAS
 +RAS0*/RASENB*RFDBL                    only addressed bank gets
 +RAS0*/RASENBL*/Q4*RFDBL               RAS
 +RAS0*/RASENBL*Q3*RFDBL
 +/RFDBL*/RASENB*/Q2                    ) refresh RASs are
 +/RFDBL*/RASENBL*Q1                    ) staggered to reduce
 +/RFDBL*/RASENBL*Q2                    ) current spike


RAS1 =
 /RASENB*/Q2*A20*/A21*RFDBL             memory bank 1 RAS
 +RAS1*/RASENB*RFDBL
 +RAS1*/RASENBL*/Q4*RFDBL
 +RAS1*/RASENBL*Q3*RFDBL
 +/RFDBL*RAS0*/Q4                       )
 +/RFDBL*/RASENBL*Q3                    ) staggered refresh RAS
 +/RFDB*/RFLATB*Q4                      )
```

```
RAS2 =
 /RASENB*/Q2*/A20*A21*RFDBL          memory bank 2 RAS
 +RAS2*/RASENB*RFDBL
 +RAS2*/RASENBL*/Q4*RFDBL
 +RAS2*/RASENBL*Q3*RFDBL
 +/RFDBL*/RASENBL*Q3                 )
 +/RFDB*/RFLATB*/Q2                  ) staggered refresh RAS
 +/RFDBL*/RASENBL*/RFLATB*Q1         )

RAS3 =
 /RASENB*/Q2*A20*A21*RFDBL           memory bank 3 RAS
 +RAS3*/RASENB*RFDBL
 +RAS3*/RASENBL*/Q4*RFDBL
 +RAS3*/RASENBL*Q3*RFDBL
 +/RFDBL*RAS2*/RFLATB*/Q3            )
 +/RFDBL*RASENBL*/RFLATB*Q2          ) staggered refresh RAS
 +/RFDBL*RASENBL*/RFLATB*Q3          )

RFRQB =
 RFRA*RFRB                           to ensure refresh is never
                                     skipped during long
                                     sequential runs
```

5.1.3 IC 3 PAL20L8

Clock generator for 2/4 Mb machine (A2M2PCKGEN)

This PAL is configured as a four-phase overlapping clock generator in the form of a divide-by-three ring counter. Each edge of the input clock moves a pulse one step along.

Q1B..Q4B

These are the four overlapping output clocks, each at one-third of the frequency of the input clock.

```
Q1B =
 Q1B*/CK                              closed when CK low
 +/Q2B*CK                             ) divide by three
 +/Q4B*CK                             )
 +/Q2B*Q1B                            ) prevent race hazard
 +/Q4B*Q1B                            )


Q2B =
 Q1B*/CK                              closed when CK high
 +Q2B*CK                              shift when CK low
 +Q1B*Q2B                             prevent race


Q3B =
 Q3B*/CK                              closed when CK low
 +Q2B*CK                              shift when CK high
 +Q2B*Q3B                             prevent race


Q4B =
 Q3B*/CK                              closed when CK high
 +Q4B*CK                              shift when CK low
 +Q3B*Q4B                             prevent race
```

5.1.4 IC 9 PAL20R8

Sequence generator for 2/4 MByte machine (A2M2SEQ)

This PAL generates the enables for the processor clocks and the memory strobes which are then converted into precisely timed signals by A2M2TIM1 and A2M2TIM2.

RFRQL
A synchronising latch for the refresh request signal.

PHWTB
A delayed copy of the wait state request.

RFLAT
Latches the refresh request to ensure one refresh operation for every cycle of the refresh request.

PB
Memory cycle pending. The processor MRE signal is pipelined by one processor cycle. If a refresh cycle takes place the value of MREQ would be lost, so it is preserved here until needed.

PHIEN
Processor clock enable. The processor clock is stopped during refresh and stretched during non-sequencial cycles.

CASEN
RAM column address strobe enable.

RASEN
RAM row address strobe enable.

RFADD
Tells A2PTIM2 to enable the RAM refresh address buffers.

```
RFRQL :=
 RFRQ                                        synchronise RFRQ


PHWTB :=
 /WAIT                                       ph 2 stretch


RFLAT :=
 RFLAT*RFRQL                                 latch refresh request
 +RFADD*RASEN                                until refresh started

PB :=
 MREQB*PB                                    internal cycle enable
 +/PHIEN*/RFADD
 +/RFADD*SEQ*RT*/A24*/A25*PHIEN*/A2          preserve pending
 +/RFADD*SEQ*RT*/A24*/A25*PHIEN*/A3          memory request
 +/RFADD*SEQ*RT*/A24*/A25*PHIEN*/A4          until ph 2
 +/RFADD*SEQ*RT*/A24*/A25*PHIEN*/A5
 +/RFADD*SEQ*RT*/A24*/A25*PHIEN*/A6


PHIEN :=
 /RFADD*SEQ*RT*/A24*/A25*PHIEN*/A2           enable ph 2 when
 +/RFADD*SEQ*RT*/A24*/A25*PHIEN*/A3          not a refresh
 +/RFADD*SEQ*RT*/A24*/A25*PHIEN*/A4          cycle and
 +/RFADD*SEQ*RT*/A24*/A25*PHIEN*/A5          sequential and
 +/RFADD*SEQ*RT*/A24*/A25*PHIEN*/A6          not at 32-word
 +MREQB*PB                                   boundary or
 +/RFADD*/PHIEN                              internal cycle
 +WAIT                                       or non-sequential
                                             or stretching


CASEN :=
 /RFADD*SEQ*RT*/A24*/A25*PHIEN*/A2           enable CAS when
 +/RFADD*SEQ*RT*/A24*/A25*PHIEN*/A3          RAM and ph 2
 +/RFADD*SEQ*RT*/A24*/A25*PHIEN*/A4          enabled
 +/RFADD*SEQ*RT*/A24*/A25*PHIEN*/A5
 +/RFADD*SEQ*RT*/A24*/A25*PHIEN*/A6
 +/RFADD*/PHIEN*RT*/A24*/A25
```

```
RASEN :=
 /RFADD*SEQ*RT*/A24*/A25*PHIEN*/A2          enable RAS as CAS
 +/RFADD*SEQ*RT*/A24*/A25*PHIEN*/A3         enable plus for
 +/RFADD*SEQ*RT*/A24*/A25*PHIEN*/A4         refresh
 +/RFADD*SEQ*RT*/A24*/A25*PHIEN*/A5
 +/RFADD*SEQ*RT*/A24*/A25*PHIEN*/A6
 +/RFADD*/PHIEN*RT*/A24*/A25
 +RFADD*/RASEN*/RFLAT

RFADD :=
 RFADD*/RFLAT                               refresh when
                                            requested
 +PHIEN*RFRQL*/WAIT*/RFLAT*/RT              and not stretching
 +PHIEN*RFRQL*/WAIT*/RFLAT*/SEQ             and not sequential
 +PHIEN*RFRQL*/WAIT*/RFLAT                  if RAM and ph 2
 *A2*A3*A4*A5*A6                            enabled
 +PHIEN*RFRQL*/WAIT*/RFLAT*A24
 +PHIEN*RFRQL*/WAIT*/RFLAT*A25
```

5.1.5 IC 18 PAL20R8

ROM State machine 2/4 MByte (A2M2PROM)

The ROM state machine. This PAL is responsible for controlling ROM and TUBE accesses. One byte wide ROM is used on this board and four accesses are made each time the ROM is addressed to construct a 32-bit word. This PAL counts the bytes out of ROM registers and requests wait states while this takes place.

RDS/WDS
Read and write strobe outputs to control TUBE, with appropriate wait states and address decoding.

RSTLABT
Latches a reset signal to enable the reset vector to be fetched from ROM. Usually low addresses enable RAM, but it is clearly important that address 0 (the reset vector) is fetched from ROM, especially on power-up.

WAITB
Controls wait states for TUBE and ROM accesses.

RB0B, RB1B
Low-order ROM addresses. These count the four bytes out of the ROM to form a word.

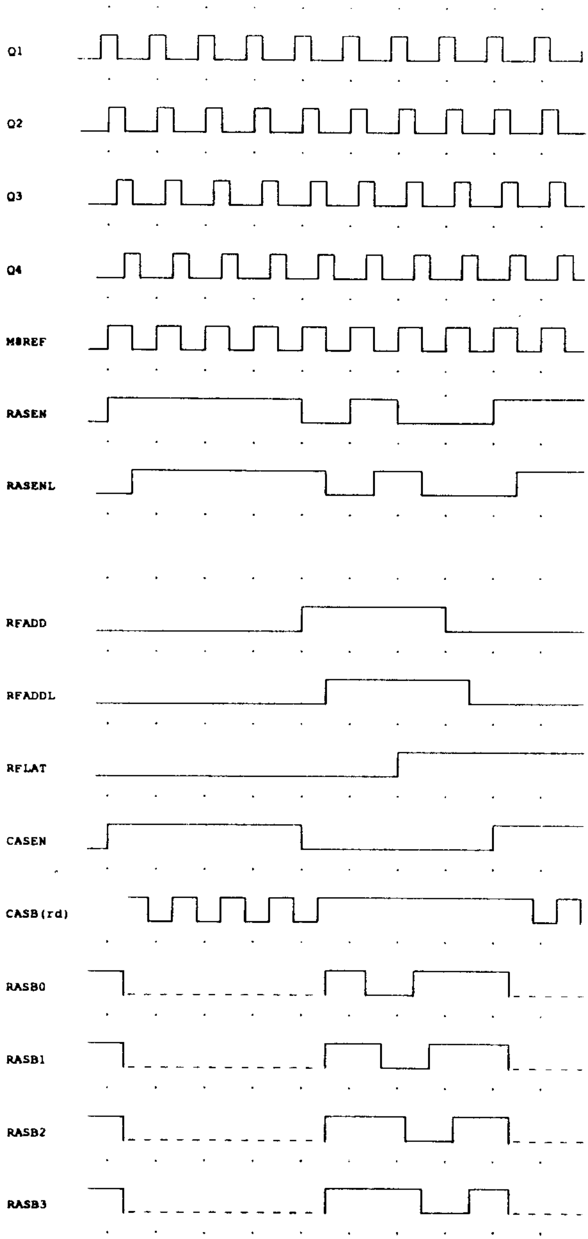
RCKB
Shifts bytes out of the ROM across the three ROM latches.

RCS
The ROM chip select.

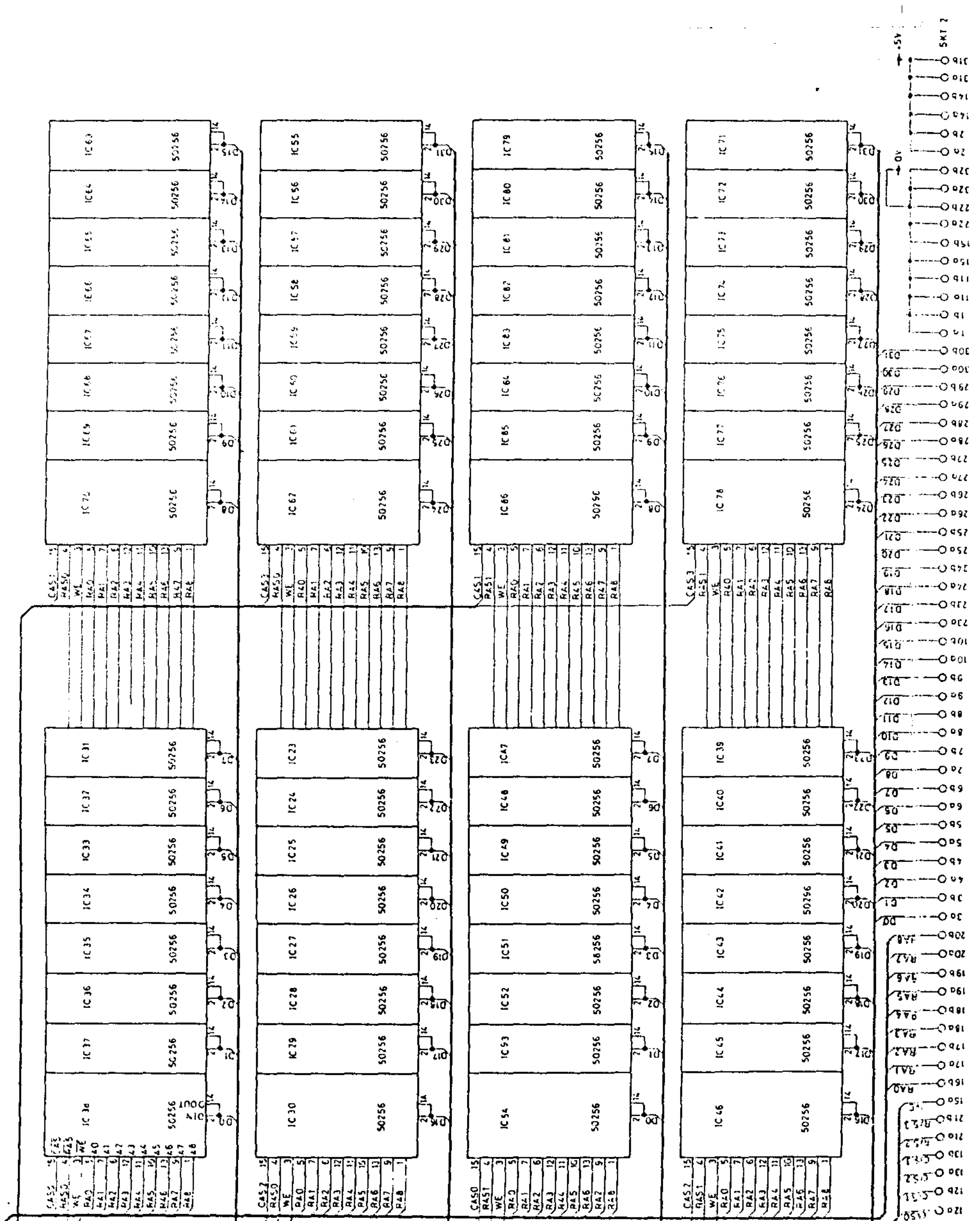
```
RDS :=
 /WDS*/WAITB*/A25*A24*/WRL        TUBE read
 +/WDS*/WAITB*/A25*A24*RDS        stretch

WDS :=
 /RDS*/WAITB*/A25*A24*WRL         TUBE write
 +/RDS*/WAITB*/A25*A24*WDS        stretch
```

```
RSTLATB :=
 /RSTB                          set on reset
 +RSTLATB*/WRL                  cleared by first write

WAITB :=
 WAITB*PHI1B                    start wait in ph1
 +/A24*/RSTLATB                 do not wait for RAM
 +/A25*A24*/RCKB                end of TUBE wait cycle
 +/RB0B*/RB1B*/RCKB             end of ROM wait cycle

RB0B :=
 /WAITB*RB0B*/RCKB              hold value when RCKB low
 +/WAITB*/RB0B*RCKB             toggle when RCKB high
                                (=RCK/2)

RCS :=
 A24*A25                        normal ROM select
 +/A24*/A25*RSTLATB             to fetch reset vector

RB1B :=
 /WAITB*/RB1B*/RB0B*RCKB        toggle when RCKB high
                                & RBOB low
 +/WAITB*RB1B*RB0B              hold value when RBOB high
 +/WAITB*RB1B*/RCKB             hold value when RBOB low
                                (=RCK/4)

RCKB :=
 /RCKB                          toggle
 +WAITB
```

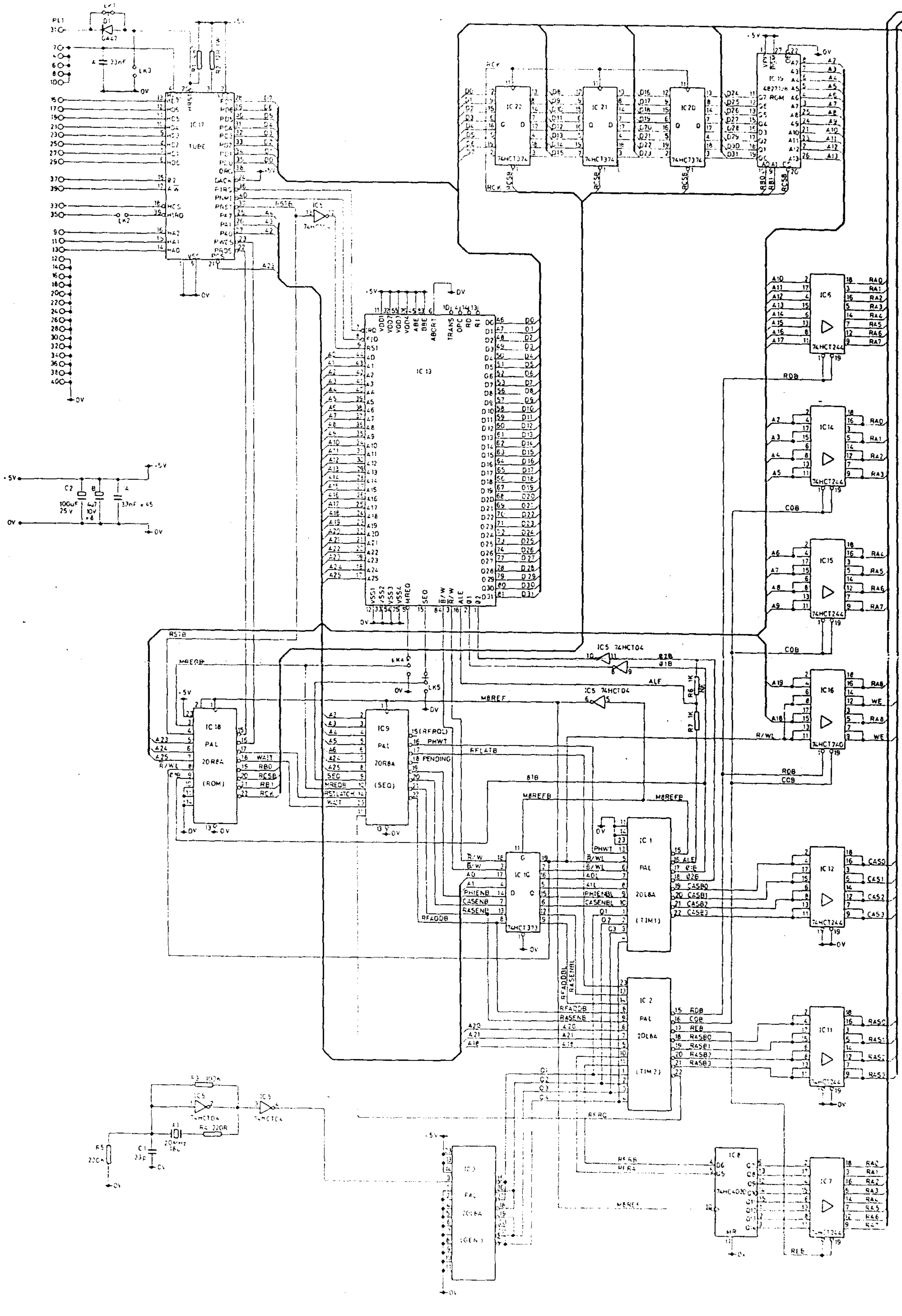
5.2 Timing diagrams

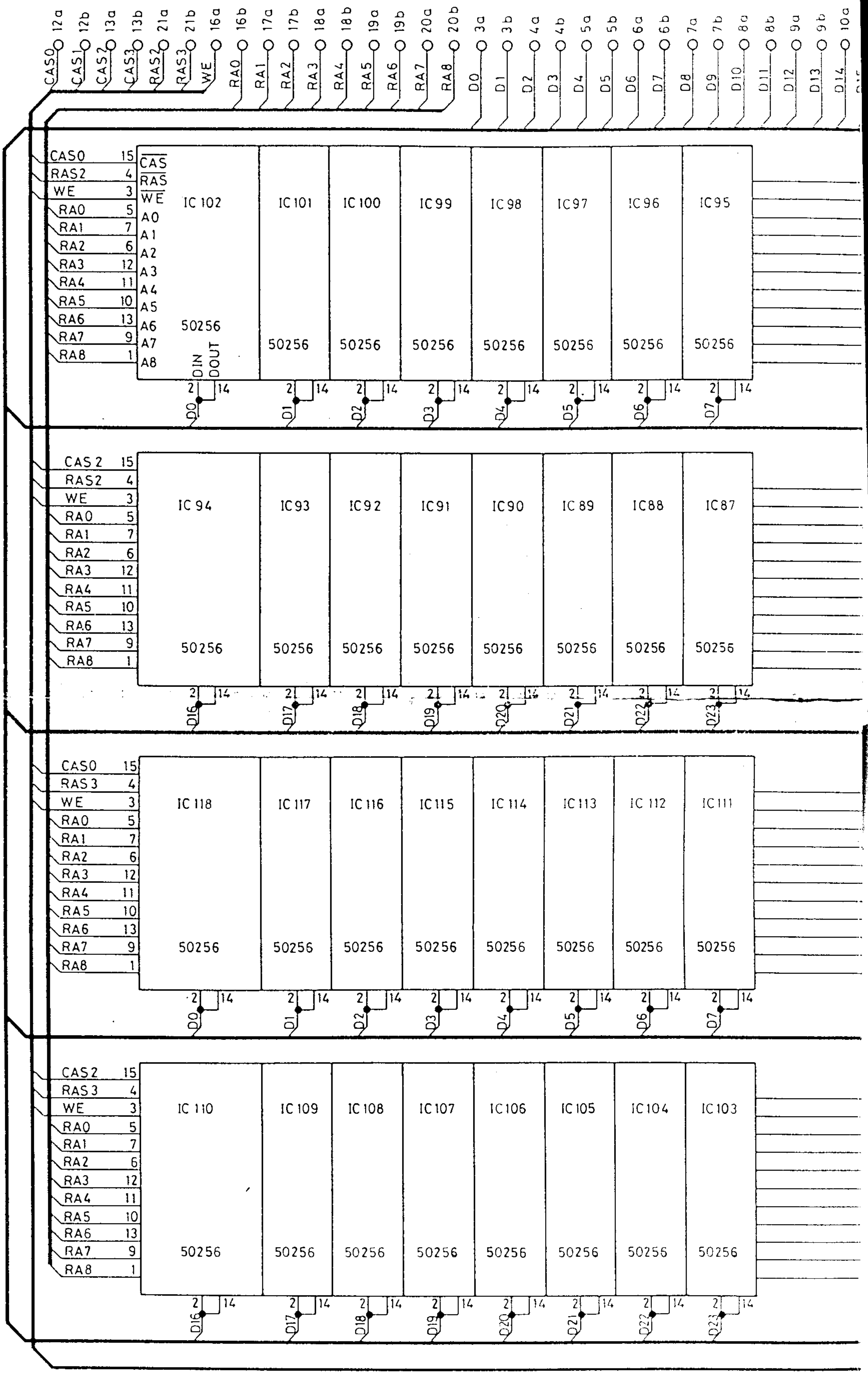


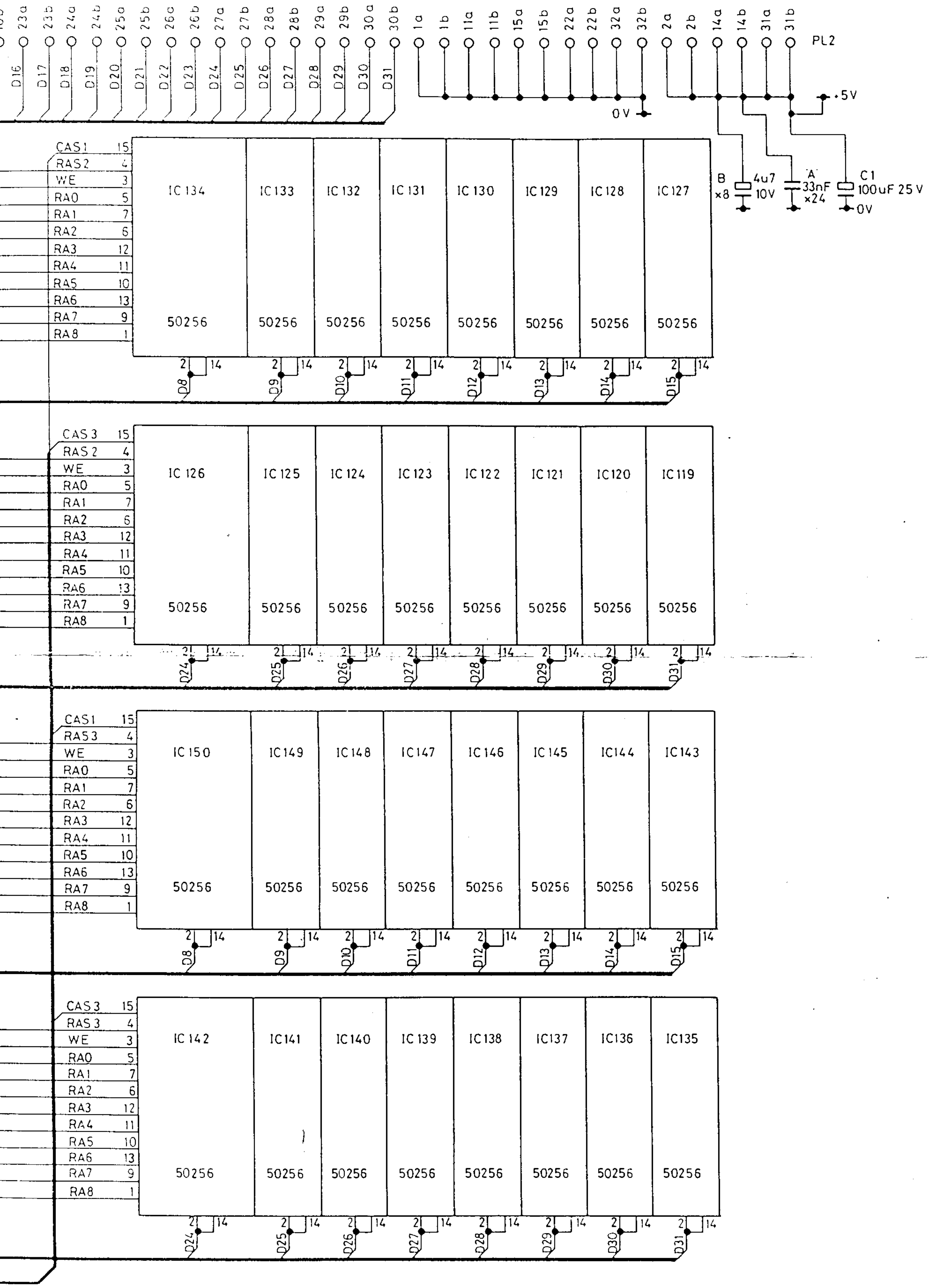
5.3 ARM co-processor circuit diagram



IC	0V	+5V	TYPE No
1	12	24	20L8A
2	12	24	20L8A
3	12	24	20L8A
4	DOES NOT EXIST		
5	7	14	74HCT04
6	10	20	74HCT244
7	10	20	74HCT244
8	8	16	74HC4020
9	12	24	20R8A
10	10	20	74HCT373
11	10	20	74HCT244
12	10	20	74HCT244
13			SEE DIAGRAM
14	10	20	74HCT244
15	10	20	74HCT244
16	10	20	74HCT240
17	1,5	2	TUBE
18	12	24	20R8A
19	14	28	27128 ROM
20	10	20	74HCT374
21	10	20	74HCT374
22	10	20	74HCT374
23-86	16	8	50256


IC 17
TUBE
PL1
D1
OA47
LK1
LK2
LK3
33nF
IC 22
IC 21
IC 20
74HC374
IC 19
4827128 ROM
RCK
RCSB
RB0
RB1
IC 13
IC 5
IC 6
IC 14
IC 15
IC 16
IC 11
IC 12
IC 7
74HCT244
74HCT740
C2
100uF
25 V
4u7
10V
33nF x 45
+5V
0V
IC 18
PAL
20R8A
(ROM)
IC 9
PAL
20R8A
(SEQ)
IC 10
74HCT373
IC 1
PAL
20L8A
(TIM1)
IC 2
PAL
20L84
(TIM2)
IC 3
PAL
20L8A
(GEN)
IC 8
74HC4020
IC5 74HCT04
R6 1K
R7 1K
R3 100K
R4 220R
R5 220R
X1 20MHz
C1 33p
LK4
LK5
MBREF
MBREFB
RFLATB
RS1B
MREQB
ALE
PHWT
PENDING
WAIT
ROB
COB
CCB
RAS
CAS


CAS0 12a
CAS1 12b
CAS2 13a
CAS3 13b
RAS2 21a
RAS3 21b
WE 16a
RA0 16b
RA1 17a
RA2 17b
RA3 18a
RA4 18b
RA5 19a
RA6 19b
RA7 20a
RA8 20b
D0 3a
D1 3b
D2 4a
D3 4b
D4 5a
D5 5b
D6 6a
D7 6b
D8 7a
D9 7b
D10 8a
D11 8b
D12 9a
D13 9b
D14 10a
CAS0 15
RAS2 4
WE 3
RA0 5
RA1 7
RA2 6
RA3 12
RA4 11
RA5 10
RA6 13
RA7 9
RA8 1
CAS
RAS
WE
A0
A1
A2
A3
A4
A5
A6
A7
A8
DIN
DOUT
IC 102
IC 101
IC 100
IC 99
IC 98
IC 97
IC 96
IC 95
50256
2
14
D0
D1
D2
D3
D4
D5
D6
D7
CAS 2 15
RAS2 4
WE 3
IC 94
IC 93
IC 92
IC 91
IC 90
IC 89
IC 88
IC 87
D16
D17
D18
D19
D20
D21
D22
D23
CAS0 15
RAS 3 4
WE 3
IC 118
IC 117
IC 116
IC 115
IC 114
IC 113
IC 112
IC 111
CAS2 15
RAS 3 4
WE 3
IC 110
IC 109
IC 108
IC 107
IC 106
IC 105
IC 104
IC 103


PL2
23a 23b 24a 24b 25a 25b 26a 26b 27a 27b 28a 28b 29a 29b 30a 30b
1a 1b 11a 11b 15a 15b 22a 22b 32a 32b
2a 2b 14a 14b 31a 31b
D16 D17 D18 D19 D20 D21 D22 D23 D24 D25 D26 D27 D28 D29 D30 D31
0V
+5V
B
×8
4u7
10V
'A'
33nF
×24
C1
100uF 25V
0V
CAS1 15
RAS2 4
WE 3
RA0 5
RA1 7
RA2 6
RA3 12
RA4 11
RA5 10
RA6 13
RA7 9
RA8 1
IC134 IC133 IC132 IC131 IC130 IC129 IC128 IC127
50256
2
14
D8 D9 D10 D11 D12 D13 D14 D15
CAS3 15
RAS2 4
IC126 IC125 IC124 IC123 IC122 IC121 IC120 IC119
D24 D25 D26 D27 D28 D29 D30 D31
CAS1 15
RAS3 4
IC150 IC149 IC148 IC147 IC146 IC145 IC144 IC143
D8 D9 D10 D11 D12 D13 D14 D15
CAS3 15
RAS3 4
IC142 IC141 IC140 IC139 IC138 IC137 IC136 IC135
D24 D25 D26 D27 D28 D29 D30 D31

6. Appendix A

6.1 ARM instruction set

There are 14 instructions, which are determined by the bit-pattern in B24-B27, divided into 5 classes.

B27	B26	B25	B24	mnemonics	instruction type
1	0	1	0	B	BRANCH
1	0	1	1	BL	BRANCH WITH LINK
0	0	0	X	various	DATA
					PROCESSING
0	0	1	X	various	GROUP
0	1	0	0	LDR/STR	
0	1	0	1	LDR/STR	SINGLE
					DATA TRANSFER
0	1	1	0	LDR/STR	GROUP
0	1	1	1	LDR/STR	
1	0	0	0	LDM/STM	BLOCK DATA
					TRANSFER post inc/dec
1	0	0	1	LDM/STM	BLOCK DATA
					TRANSFER pre inc/dec
1	1	1	1	SWI	SUPERVISOR CALL
1	1	0	0		
					reserved
1	1	0	1		for future
					expansion
1	1	1	0		

The basic instruction set is expanded by altering the pattern of the remaining 28 bits

7. Appendix B

7.1 Document references

Related documents:

- *ARM software reference manual*
- *ARM ASSEMBLER reference manual*
- *TWIN reference manual*
- *BBC BASIC reference manual*
- *LISP reference manual*
- *PROLOG reference manual*
- *FORTRAN reference manual*
- *C reference manual*
- *ARM system user guide.*

Acorn

The choice of experience.