ARM hardware
reference manual

ARM Evaluation System

Acorn
The choice of experience.

Acorn OEM Products
ARM hardware
© Copyright Acorn Computers Limited 1986

Neither the whole nor any part of the information contained in, or the product described in, this manual may be adapted or reproduced in any material form except with the prior written permission of the copyright holder. The only exceptions are as provided for by the Copyright (photocopying) Act, or for the purpose of review, or in order for the software herein to be entered into a computer for the sole use of the owner of this book.

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

All maintenance and service on the product must be carried out by Acorn Computers. Acorn Computers can accept no liability whatsoever for any loss, indirect or consequential damages, even if Acorn has been advised of the possibility of such damage or even if caused by service or maintenance by unauthorized personnel. This manual is intended only to assist the reader in the use of the product, and therefore Acorn Computers shall not be liable for any loss or damage whatsoever arising from the use of any information or particulars in, or any error or omission in, this manual, or any incorrect use of the product.

Econet® and The Tube® are registered trademarks of Acorn Computers Limited.
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 Features
   1.2 Performance

2. Standard specifications
   2.1 Physical description
   2.2 Power requirements
   2.3 Temperatures

3. Functional description
   3.1 ARM block diagram
   3.2 Pin connections, pins 1 to 42
   3.3 ARM element functions
   3.4 ARM operational description

4. Signal description
   4.1 D.C. characteristics
   4.2 Absolute maximum ratings
   4.3 Signal definitions
   4.4 Timing information, signal dwell times
   4.5 Timing waveforms

5. The ARM co-processor
   5.1 Description of the ICs
      5.1.1 IC 1 PAL20L8
      5.1.2 IC 2 PAL20L8
      5.1.3 IC 3 PAL20L8
      5.1.4 IC 9 PAL20R8
      5.1.5 IC 18 PAL20R8
   5.2 Timing diagrams
   5.3 ARM co-processor circuit diagram

6. Appendix A
   6.1 ARM instruction set

7. Appendix B
   7.1 Document references
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

<table>
<thead>
<tr>
<th>absolute maximum rating</th>
<th>nominal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vcc (supply voltage with respect to Vss)</td>
<td>+6.0 V d.c.*</td>
</tr>
</tbody>
</table>

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

![ARM floorplan diagram]

ARM floorplan
### 3.2 Pin connections, pins 1 to 42

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

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

<table>
<thead>
<tr>
<th>element</th>
<th>function</th>
</tr>
</thead>
<tbody>
<tr>
<td>alu</td>
<td>arithmetic logic unit</td>
</tr>
<tr>
<td>registers</td>
<td>a bank of 25 32-bit registers</td>
</tr>
<tr>
<td>shift</td>
<td>a 32-bit barrel shifter to assist arithmetic and logical register operations with a rotate capability</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>name</th>
<th>min</th>
<th>typical</th>
<th>max</th>
<th>unit</th>
<th>conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vil</td>
<td>-0.30</td>
<td>+0.80</td>
<td></td>
<td>V</td>
<td>except clock</td>
</tr>
<tr>
<td>Vih</td>
<td>+2.40</td>
<td>Vcc</td>
<td></td>
<td>V</td>
<td>except clock</td>
</tr>
<tr>
<td>Vceil</td>
<td></td>
<td>+0.30</td>
<td></td>
<td>V</td>
<td>clock</td>
</tr>
<tr>
<td>Vcih</td>
<td>Vcc</td>
<td>Vcc</td>
<td></td>
<td>V</td>
<td>clock</td>
</tr>
<tr>
<td>Vol</td>
<td></td>
<td>+0.50</td>
<td></td>
<td>V</td>
<td>Iol = -3.60 mA</td>
</tr>
<tr>
<td>Voh</td>
<td>Vcc</td>
<td></td>
<td></td>
<td>V</td>
<td>Ioh = +3.00 mA</td>
</tr>
<tr>
<td>Vcc</td>
<td>+4.75</td>
<td>+5.00</td>
<td>+5.25</td>
<td>V</td>
<td>measured at Vcc = 5.00 V</td>
</tr>
<tr>
<td>Icc</td>
<td>20.00</td>
<td>mA</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

symbols

Vil  signal input low voltage
Vih  signal input high voltage
Vceil 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 (firq)

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.

Signal description

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.

<table>
<thead>
<tr>
<th>m1</th>
<th>m0</th>
<th>mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>supervisor</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>irq</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>fiq</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>user</td>
</tr>
</tbody>
</table>

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 incremeneter, 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

<table>
<thead>
<tr>
<th>signal name</th>
<th>measured period (nsec)</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tv</td>
<td>0</td>
<td>measured at 2 V level - no clock overlap</td>
</tr>
<tr>
<td>Tckl</td>
<td>70</td>
<td>phase 2 clock low period</td>
</tr>
<tr>
<td>Tckh</td>
<td>55</td>
<td>phase 2 clock high period</td>
</tr>
<tr>
<td>Tckl+Tckh</td>
<td>$10^9 \times 1$</td>
<td>(estimate)</td>
</tr>
<tr>
<td>Taout</td>
<td>0</td>
<td>address out delay time</td>
</tr>
<tr>
<td>Tah</td>
<td>10</td>
<td>address hold time</td>
</tr>
<tr>
<td>Trws</td>
<td>0</td>
<td>read/write set up</td>
</tr>
<tr>
<td>Trwh</td>
<td>10</td>
<td>read/write hold time</td>
</tr>
<tr>
<td>Tdout</td>
<td>50</td>
<td>data out delay time</td>
</tr>
<tr>
<td>Tdoh</td>
<td>10</td>
<td>data out hold time</td>
</tr>
<tr>
<td>Tdis</td>
<td>5</td>
<td>data in set time</td>
</tr>
<tr>
<td>Tdih</td>
<td>10</td>
<td>data in hold time</td>
</tr>
<tr>
<td>Tdbe</td>
<td>30</td>
<td>data bus enable to data bus valid</td>
</tr>
<tr>
<td>Tdbz</td>
<td>35</td>
<td>data bus enable to data bus high impedance</td>
</tr>
<tr>
<td>Tabs</td>
<td>35</td>
<td>abort set-up time</td>
</tr>
<tr>
<td>Tabh</td>
<td>0</td>
<td>abort hold time</td>
</tr>
<tr>
<td>Tseqd</td>
<td>55</td>
<td>sequential indicated delay</td>
</tr>
<tr>
<td>Tseqh</td>
<td>10</td>
<td>sequential indicated hold</td>
</tr>
</tbody>
</table>

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.
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  
+Q2  
synthesize 8 MHz square-wave

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

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

PHI2 =  
PHWT*/PHI1  
+/PHIENBL*/Q2*/PHI1  
+/PHIENBL*/Q1*/PHI1  
stretch phase 2 high ) time phase 2 edges )
CAS0 =
BWL*/CASENBL*Q1*/WRL
access +BWL*/CASENBL*Q2
+BWL*/CASENBL*Q3*/WRL
+/A0L*/A1L*/CASENBL*Q1*/WRL
+/A0L*/A1L*/CASENBL*Q2
+/A0L*/A1L*/CASENBL*Q3*/WRL
start read early - word
end read late - word access

CAS1 =
BWL*/CASENBL*Q1*/WRL
+/BWL*/CASENBL*Q2
+/BWL*/CASENBL*Q3*/WRL
+/A0L*/A1L*/CASENBL*Q1*/WRL
+/A0L*/A1L*/CASENBL*Q2
+/A0L*/A1L*/CASENBL*Q3*/WRL
word access
word access
word access

CAS2 =
BWL*/CASENBL*Q1*/WRL
+/BWL*/CASENBL*Q2
+/BWL*/CASENBL*Q3*/WRL
+/A0L*/A1L*/CASENBL*Q1*/WRL
+/A0L*/A1L*/CASENBL*Q2
+/A0L*/A1L*/CASENBL*Q3*/WRL
word access
word access
word access

CAS3 =
BWL*/CASENBL*Q1*/WRL
+/BWL*/CASENBL*Q2
+/BWL*/CASENBL*Q3*/WRL
+/A0L*/A1L*/CASENBL*Q1*/WRL
+/A0L*/A1L*/CASENBL*Q2
+/A0L*/A1L*/CASENBL*Q3*/WRL
word access
word access
word access

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
+RASENB*/Q1*RFDBL*RFDB
+RASENB*/Q2*RFDBL*RFDB
+RASENB*/RASENB*RFDBL*RFDB
row address enable

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

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

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

RAS1 =
/RASENB*/Q2*/A20*/A21*RFDBL
+RAS1*/RASENB*/RFDBL
+RAS1*/RASENB*/Q4*RFDBL
+RAS1*/RASENB*/Q3*RFDBL
+RFDBL*/RAS0*/Q4
+RFDBL*/RASENB*/Q3
+RFDBL*/RFDBL*/Q4
memory bank 1 RAS

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

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

RFRQB =
RFRA*/FRFB
to ensure refresh is never
skipped during long
sequential runs

28 ARM hardware

29 ARM hardware
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.

\[
\begin{align*}
Q1B &= \frac{Q1B \cdot \overline{CK}}{3} \\
+Q2B &= \frac{Q1B \cdot \overline{CK}}{3} \\
+Q4B &= \frac{Q1B \cdot \overline{Q1B}}{3} \\
+Q4B &= \frac{Q1B \cdot Q1B}{3}
\end{align*}
\]

Q2B = closed when CK low
+Q2B = divide by three
+Q1B = prevent race hazard
+Q4B = prevent race

Q3B = closed when CK high
+Q2B = shift when CK low
+Q1B = prevent race

Q4B = closed when CK high
+Q4B = shift when CK low
+Q3B = 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-sequential cycles.

CASEN
RAM column address strobe enable.

RASEN
RAM row address strobe enable.

RFADD
Tells A2PTIM2 to enable the RAM refresh address buffers.
Chapter 5

RFRQL :=
RFRQ

synchronise RFRQ

PHWTB :=
/WAIT

ph 2 stretch

RFLAT :=
RFLAT*RFRQL
+RFADD*RASEN

latch refresh request
until refresh started

PB :=
MREQB*PB
+/PHIEN*/RFADD
+/RFADD*SEQ*RT*/A24*/A25*/PHIEN*/A2
+/RFADD*SEQ*RT*/A24*/A25*/PHIEN*/A3
+/RFADD*SEQ*RT*/A24*/A25*/PHIEN*/A4
+/RFADD*SEQ*RT*/A24*/A25*/PHIEN*/A5
+/RFADD*SEQ*RT*/A24*/A25*/PHIEN*/A6

internal cycle enable
preserve pending
memory request
until ph 2

PHIEN :=
/RFADD*SEQ*RT*/A24*/A25*/PHIEN*/A2
+/RFADD*SEQ*RT*/A24*/A25*/PHIEN*/A3
+/RFADD*SEQ*RT*/A24*/A25*/PHIEN*/A4
+/RFADD*SEQ*RT*/A24*/A25*/PHIEN*/A5
+/RFADD*SEQ*RT*/A24*/A25*/PHIEN*/A6
+MREQB*PB
+RFADD*/PHIEN
+WAIT

enable ph 2 when
not a refresh
cycle and
sequential and
not at 32-word
boundary or
internal cycle
or non-sequential
or stretching

RASEN :=
/RFADD*SEQ*RT*/A24*/A25*/PHIEN*/A2
+/RFADD*SEQ*RT*/A24*/A25*/PHIEN*/A3
+/RFADD*SEQ*RT*/A24*/A25*/PHIEN*/A4
+/RFADD*SEQ*RT*/A24*/A25*/PHIEN*/A5
+/RFADD*SEQ*RT*/A24*/A25*/PHIEN*/A6
+/RFADD*/PHIEN*/RT*/A24*/A25
+RFADD*/RASEN*/RFLAT

enable RAS as CAS
enable plus for
refresh

RFADD :=
RFADD*/RFLAT

refresh when
requested
and not stretching
and not sequential
if RAM and ph 2
enabled

+/PHIEN*RFRL*/WAIT*/RFLAT*/RT
+/PHIEN*RFRL*/WAIT*/RFLAT*/SEQ
+/PHIEN*RFRL*/WAIT*/RFLAT
*A2*A3*A4*A5*A6
+/PHIEN*RFRL*/WAIT*/RFLAT*A24
+/PHIEN*RFRL*/WAIT*/RFLAT*A25
Chapter 5

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.

RSTLABT :=
/RSTB
+RSTLABT*/WRL

WAITB :=
WAITB*P1&B
+/A24*/RSTLABT
+/A25*A24*/RCKB
+/RB0B*/RB1B*/RCKB

set on reset

cleared by first write

start wait in phi1

do not wait for RAM

end of TUBE wait cycle

end of ROM wait cycle

RB0B :=
/WAITB*RB0B*/RCKB
+/WAITB*/RB0B*RCKB

hold value when RCKB low

toggle when RCKB high

(=RCK/2)

RCS :=
A24*A25
+/A24*/A25*RSTLABT

normal ROM select
to fetch reset vector

RB1B :=
/WAITB*/RB1B*/RB0B*RCKB
+/WAITB*RB1B*RB0B
+/WAITB*RB1B*/RCKB

toggle when RCKB high & RB0B low

hold value when RB0B high

hold value when RB0B low

(=RCK/4)

RCKB :=
/RCKB
+/WAITB

toggle

RDS :=
/WDS*/WAITB*/A25*A24*/WRL
+/WDS*/WAITB*/A25*A24*RDS

TUBE read

stretch

WDS :=
/RDS*/WAITB*/A25*A24*WRL
+/RDS*/WAITB*/A25*A24*WDS

TUBE write

stretch
5.2 Timing diagrams
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.

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

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.