CMU-11 engineering documentation

Samuel H. Fuller
Carnegie Mellon University

T. M. McWilliams

W. H. Sherwood

Follow this and additional works at: http://repository.cmu.edu/compsci
NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:
The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.
ABSTRACT

The CMU-11 is a microprogrammable processor built with the Intel 3000 microcomputer set that emulates the PDP-11 architecture. In addition, it has been designed to provide full Unibus support. The enclosed documentation gives the details of the CMU-11 design. This documentation has been generated in conjunction with the Stanford Drawing System, the SAGE simulator, and the Intel 3000 microassembler. Those hoping to do any further development of the CMU-11 design are encouraged to also use these design aids and all of the CMU-11 design information shown here (and other information such as ROM contents and wirelists) are available on magnetic tape. See the following report for an introductory discussion and evaluation of the CMU-11:

CONTENTS

1. CMU-11 Register-Transfer Level Block Diagram ........ 1
2. Package Placement .................................... 2
3. Logic Drawings ....................................... 3
4. Microprogram ......................................... 36
COMPUTER SCIENCE ENGINEERING LAB

11 INSTRUCTION REGISTER

PDP-11 USING THE INTEL 3000 MICROPROCESSOR

BY:

DATE: 18-MAY-75 18:35

PITTSBURGH, PENNSYLVANIA
<table>
<thead>
<tr>
<th>Function</th>
<th>Logic Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock 2</td>
<td></td>
</tr>
<tr>
<td>Init</td>
<td></td>
</tr>
<tr>
<td>SSB1.1</td>
<td></td>
</tr>
<tr>
<td>Pulse Set MTH1</td>
<td></td>
</tr>
<tr>
<td>Init 2.1</td>
<td></td>
</tr>
<tr>
<td>Phase</td>
<td></td>
</tr>
<tr>
<td>Single Step MTH1</td>
<td></td>
</tr>
</tbody>
</table>

**COMPUTER SCIENCE ENGINEERING LAB**

**SYNCHRONIZER REGISTER**

**PROJECT:** PDP-11 USING THE INTEL 3000 MICROPROCESSOR

**SHERWOOD**

**CHECKED BY:**

**DATE:** 30-DEC-76 22:22

**PHOTOGRAPH BY:** Busick [N210TM95]

**CENSUS-SHELDON UNIVERSITY**
COMPUTER SCIENCE ENGINEERING LAB

PROCESSOR CLOCK AND INITIALIZATION

PROJECT:

DATE: 27 JUN 75 17:30

SHERWOOD

PITTSBURGH, PENNSYLVANIA 15233
STACK UNDERFLOW AND ODD ADDR ERRORS
PDP-11 USING THE INTEL 3000 MICROPROCESSOR

CLOCK 2 L

COMPUTER SCIENCE ENGINEERING LAB

SHERWOOD
COMPUTER SCIENCE ENGINEERING LAB

PROGRAM STATUS REGISTER

PDP-11 USING THE INTEL 3000 MICROPROCESSOR

SHERWOOD

30-DEC-75 21:55

PSW1CN210TM05

COMPUTER SCIENCE ENGINEERING LAB

PROGRAM STATUS REGISTER

PDP-11 USING THE INTEL 3000 MICROPROCESSOR

SHERWOOD

30-DEC-75 21:55

PSW1CN210TM05
SET SOURCE SIGN

CLOCK 2 L

SET DESTINATION SIGN

NEW N L

RETURN NEW C

NEW H

PS 1

U CONTROL

V CONTROL 1

V CONTROL 2

COMPUTER SCIENCE ENGINEERING LAB
OVERFLOW LOGIC
PDP-11 USING THE INTEL 3000 MICROPROCESSOR

SHREWS

04-JAN-76 04:57

PITTSBURGH, PENNSYLVANIA 15213
ASSEMBLY OF MICRO-DAT ON 6-JAN-76 AT 10:48

ADDRESS 000000 000011 000111 000111 000112 000112 000112 000112
JMED 000012 000013 000014 000015 000016 000017 000018 000019
OPCD 000020 000021 000022 000023 000024 000025 000026 000027
CPCD 000028 000029 000030 000031 000032 000033 000034 000035
PLA KURGIMULT 000036

MICRO CODE TO MAKE A PDP-11/40 OUT OF THE INTEL 3000:
MICRO PROCESSOR CHIPS:

PROGRAMMING CONVENTIONS USED IN CODE:

1. ZERO IS KEPT IN R0 SO THAT TRAP ADDRESSES CAN EASILY BE
   PUT IN IT TO CALL THE TRAP SEQUENCE. THE TRAP SEQUENCE
   PUTS ZERO BACK IN R0 WHEN IT IS DONE.
2. THE 2 BIT IS USED TO DETECT DOUBLE BUS ERRORS AND
   TO INDICATE THAT THE PROCESSOR IS IN CONSOLE MODE.
   IF IT IS SET AND A TRAP OCCURS, THE MICROPROCESSOR
   WILL JUMP TO THE CONSOLE CODE.

FIELD PLA 3:0, PLA1+2, PLA4+5, PLA6+7:

FIELD KUB 1:1, KUB+0, KUB+1:
NOTE: ALL K LINES ARE INVERTED.

FIELD RG1 Z, B, R1, R2, R3, R4, R5, R6:

KTY=2, SETS1=2, SETS2=2
SCLR=1, SSMID1+1, SSMID2=0
SNOR=1, SSMID1=1, SSMID2=0

WORD PROCESSOR C#: a608.

THE R9 IS DONE.

TRAP CALL MICROPROCESSOR MODE.

R0, R1, R2, R3, R4, R5, R6:

SAOC2=010101

S1, KTPS=181101B, KTPB=100181B

K71H=000H001B

K0, K10H=11000110B

K1, K11H=11101100B

K2, K12H=11000111B

K3, K13H=11101101B

K4, K14H=11000111B

K5, K15H=11101101B

K6, K16H=11000100B

K7, K17H=11100001B, K28=10111110B

K28+010H

K29=10111101B, K30=10111101B

K31=11110110B

K32=22222333

SETZ2+2, SETZ2+2

SCLRZ=01010110B, SSMID2=10101100B

SMMZ=00000100B, SMMZ=01000100B

SMMZ=01011110B

SMMZ=01011110B

SMMZ=01011110B

PSE3=01111100B, PSCN=00010100B

---

PUT RESULT OF LAST INSTRUCTION IN MEMORY
HOLD NEXT INSTRUCTION.

1 WORDS:

SOR 84, K91, REG, JMP FETCH 1 R(1)+4C

2 WORDS:

SOR 71, K11, K77 1 1+AC AND 377

1 WORDS:

LR 8, R2Z 1725 AC(R12)

2 WORDS:

LR 8, R1L, K8 146 AC,7+AC AND 177460

3 WORDS:

SOR 89, K19, REG, JMP FETCH 1857R1K+2C
LOAD SOURCE OPERAND INTO T. FORMAT IS SSDO

INSTRUCTION CLASS 1
REGISTER MODE=0: (R(N)

NOP PLA4,JPX 84
LDI AC,1,KA1,PLA4,JMP SRCM1

DUSE0: ILR R0,PLC,JPX DUSE0

REGISTER MODE=1: (R(N)+1

NOP PLA7,DIP,JMP DESMR

DUSE1: LMI R0,KR1,PAL,PAL1,EDIP,INCR

INSTRUCTION CLASS 2
REGISTER MODE=0: (R(N)

LMI R0,RG2,EDIP,INCR

DUSE2: LMI R0,PLC,JPX DUSE2

REGISTER MODE=2: -R(N)

DUSE3: LMI R7,1,GWDI

REGISTER MODE=3: R(N)

SRCM2: LMI R7,1,GWDI