ISTITUTO NAZIONALE DI FISICA NUCLEARE

Sezione di Torino

INFN/TC-88/12
21 Marzo 1988

D. Corsetto and Shao Beibei:

A software package for testing and debugging the fast digital data processor board

Servizio Documentazione
dei Laboratori Nazionali di Frascati

INFN - Istituto Nazionale di Fisica Nucleare
Sezione di Torino

INFN/TC-88/12
21 Marzo 1988

A SOFTWARE PACKAGE FOR TESTING AND DEBUGGING THE FAST DIGITAL DATA PROCESSOR BOARD

D.Crosetto
Istituto Nazionale di Fisica Nucleare - Sezione di Torino

Shao Beibei
Institute of High Energy Physics
Academia Sinica Beijing China

ABSTRACT

The FDDP is a modular system implemented on a VME board, consisting of two 'Logical Units' per board, which can process in parallel several digitized analog signals. Each unit has a Digital Signal Processor (DSP Texas TMS32010) which executes the majority of its instructions in 160 nsec. This software package provides three programs, which can be used to test the hardware of the FDDP board and for DSP TMS32010 software debugging.

INTRODUCTION

The Fast Digital Data Processor[1] is a VME Bus based slave module which performs fast parallel digital processing. It has been developped for the preliminary data reduction of the trigger signals from the Forward Electro Magnetic Calorimeter(FEMC) of the DELPHI detector of LEP at CERN. Additionally, the FDDP board can be regarded as a general purpose board, which performs the following tasks, working in a slave parallel mode:

Fetch 8 channels digitalized signals from the analog to digital conversion card (Anin card), which is piggy-backed on its mother board.

Process the digital data with a fast microprocessor (TMS 32010),which does 16 bits multiplication in two cycles (160 nSec. per cycle).

Store the results into the FIFO memories, readable either from front panel or VME bus.

The FDDP board consists of a VME based mother board, hosting two logical units, and two Anin cards and a FIFO card that are piggy-backed on the mother board. Each unit has 2 kilo words (16 bits) program memories, a command register, a status register, a mailbox which is used for communication between the VME bus and the TMS bus, and ancillary logic to control the different parts. An interrupt control chip MC68153 serves for the two units.

Up to 12 FDDP boards i.e. 24 units, which involve 192 chan-nel A/D converters, can be located in one VME crate. A master CPU board is used as monitor. More than one hundred integrated circuits have been used on the FDDP board. It is extremely important to have an automatic and interactive program to test the hardware of the board and an effective softtool to debug the application program used for the Digital Signal Processor.

This software package provides three useful programs for testing the FDDP board and for the DSP software debugging.

THE PROGRAM STRUCTURES

All the programs are written in MC68000 Assembly language. The cross assembler used is Lattice[3] and Quelo[4], which are based on IBM-PC and MS DOS. And the same source programs and subprograms are also assembled with MC68mil[5], which is used in VAX and microVAX. The machine codes are generated in Motorola 'S' format, for easier downloading from the host computer with error checking.

The programs are well structured by several modules. Every module does one of the testing functions. To make it easier to be read and modified, none of the modules is longer than two or three pages. The modules are all position independent and relocatable. From the main program view the modules are treated as subroutines. Parameters are transfered by stack. Standard stack initialisation and standard subprogram call and standard subprogram entry have been used in the main program and in all modules. So the modules can also be called from high level language such as Fortran, Pascal and C,according to the references [2].

These subprograms can be either assembled separately as modules, then be linked together as executable program, or as the include files to be included in the main program then to be assembled only when the main program is assembled.

The program area and the data area are separated to give the possibility of making EPROM version programs.8 kilo bytes EPROM are needed for the program and 4 kilo bytes RAM are needed for the data buffers and stack.

The input output device driver (input and output a character from serial I/O) is normally hardware dependent for different master CPU board in the VME crate. To interface the program package with different masters, the routines "inch" and "outch" in the device driver module should be modified according to the hardware of the CPU board. It may be done easily by the user referred to the files named 'iocpu1b' or 'iovme101', which are used for the CPU-1B board from the Force Company and the CPU board VME-101 from Motorola.

THE PROGRAM OPERATION

The target program assembled either from IBM-PC or from micro VAX is in the Motorola 'S' format. The communication software XTALK is useful for down loading the program from IBM-PC. The CPU board in VME crate normally has a residente program, which supports such program for down loading. No special software is needed for down loading from micro VAX.

When the program is down loaded, simply let it start working from the address $3000.

PROGRAM 1: FDDP HARDWARE TEST

This program is used for interactive hardware tests of the FDDP system. It is supposed that there are up to 24 units installed in a system.

1. Main Program

The main program does the stack initialisation and sets the global variables as described in reference [2]. Then it sets the default parameters for the start address of the first FDDP board, the number of units in the system and so on. The questions about whether those parameters should be changed are then asked on the terminal. It allows the default paramaters to be changed interactively. Then the function modules are called one by one until all the FDDP boards have been tested. Finally the program returns to the system monitor, which is defined in the device driver modules. The return address could be changed to branch to the application routine when all the system has been setup and pretested.

The TMS32010 set and reset can be tested very easily, because there is a green LED on the front panel for each FDDP unit.If there are no error messages at this stage, this means that the address bus and the bit0 of the data bus have been passed through the test,otherwise the user should check the hardware of the FDDP board following error messages such as:

```
'If "BUS TRAP ERROR" is given,
Please 1: Check address select jumpers;
       2: Check  if  D1,D2,D3,D57,D66(PAL)  are   well
          programmed;'
'TMS reset test: Type R makes green LED switch on and
 off'
'If TMS can not set or reset, check D30,D31,D57,D63.'
```

Users can select the function switch to retest the function, continue to the next function,or exit by inputting a character from keyboard following the message on the screen of the terminal.

The other subprogram or subroutine function test modules, which are called by the main program, are very similar to the above although the test procedure might be more complicated.

2. Module descriptions

1) Mailbox -- mail box test module

Mail box is used to transfer information between VME and TMS. Data are written on the VME side to be read from TMS side. Then the data are written on the TMS side read in VME side. The test performed twice to test every data bit once with 0 once with 1. If the test is passed, that means databus has been checked and the module "bustest", which test the databus lines bit by bit and points which bit on databus is always high or always low, will be skipped. Otherwise,

'Unit x mailbox TMS to VME failed, Check D12,D13.' or:
'Unit x mailbox VME to TMS failed, Check D63.'

will be displayed on the terminal. Then the data bus test will follow.

A dynamic test of the time constant of the monostable, which is used for the TMS reset, is also included in this module. Though the static test has been done before, the error message:

'Time constant on 74ls123 is too large, change less than 1 uS.'

may be displayed if it happens.

2) Memory -- Memory test module

When the program enters this module, a message:

'Memories are under test, if any error, check

D10,D11.' is displayed.

All the memories in every unit are then tested, byte by byte, by means of read the data, complement it, write it back, read it again, complement again, write it back again. The data in the memories are not changed. The byte address wiht error will be pointed out if any.

3) TMS32k -- Test if TMS32010 works

The test procedure is as follows: when TMS is reset, the machine code for TMS 32010 is loaded to memory. Then TMS32010 executes the program that modifies the content of a memory location. The VME CPU then resets TMS again and check if the TMS 32010 has been running correctly. The error message is then given if any of the TMS unit did not work properly.

4) Fifotst -- First in, first out buffer test

The FIFOs are used to transfer the data from the TMS to the VME at high speed. It works normally in the interrupt mode. As we know, if there is any problem in the interrupt control circuit, the interrupt will stop all programs, and even modify all the program codes. To avoid this in the module under test, the interrupt is disabled during testing. The FIFOs are written from TMS side and read from VME side. The interrupt will be tested afterwards.

5) Antst -- Anini board test

Each anini piggy back board provides 8 channels analog to digital convertions. A flash ADC chip TS8328 is used in every channel as 8 bits converter in 120 nsec. In this module a soft trigger has been used for testing. The data in every channel and every unit can be read out from the screen of the terminal. Notice that an electrical isolation between the analog circuit and digital circuit by means of optocoupler bas been used and a separated power supply must be connected To get the correct values the appropriate ground signal and analog signal should be connected to the front panel.

6) Intml and Intfi -- Modules for interrupt initialisation
Those two modules are used to test the interrupt control chip MC68153.
Up to 48 interrupt vectors and subroutines for 24 units are initialized in memory space. When all the initialisation has been done, the interrupt is enabled. Then if the program does not work, the MC68153 component must be checked. The chip packed in ceramic seems better than plastic one.

7) Brdtst -- Subroutine for broadcast mode test
The broadcast test is in fact only the broadcast address test. The test is by means of setting and reseting all of the TMS at the same time.

8) Sublib -- subroutine library for common use
Many commonly used routines such as ascii to binary, hexadecimal to decimal and so on are collected together and put in this module.

3. The FDDP link file

The link file named FDDP.lnk gives very clear view of the program structures. It is listed here:

```
link fddp        the main program
link mboxtst     module mailbox
link memory      module memory
link intml       module intml
link intfi       module intfi
link anintst     module antst
link fifotst     module fifotst
link tms32k      module tms32k
link sublib      module subroutine lib.
link iocpu1b     device driver module
org  $3000       program started address
```

PROGRAM 2 FDDP READ AND WRITE

This program is used to test a single unit statically. The program has the same structure as the program 1, i.e. it is relocatable and position independent, structured with many modules, down loadable from a host computer either IBM-PC or micro VAX. When the program runs, a question about which unit needs to be tested, will be asked on the screen of terminal. Then a test function select menu will be displayed on the screen. The test can be done interactively by selecting the function to be tested from the following menu:

```
                 ****FUNCTION SELECT:****

A=Anin      VME    read            D=TMS stops
B=mailBox   VME    read            E=TMS works
C=Command Reg.     read            M=mailbox  VME  write
```

```
F=Fifo     VME      read              O=cOmmand reg.  write
G=cwfifo   TMS      read              P=c&ccwfifo TMS write
S=Status   Reg.     read              V=outfifo   TMS write
T=mailbox TMS       read              X=mailboX   TMS write
I=fifo Int.ctrl     read              J=fifo int.ctrl write
U=Mbox int.ctrl     read              H=Mbox int.ctrl write
K=fifo int.ctrl     read              L=fifo int.ctrl write
Y=Mbox int.Vect     read              Z=mbox int.vect write
W=ccWfifo TMS       read              N=clear fifo
                                      Q=Quit
```

The broadcast mode test function is also included in the program by means of select unit 0 as the unit to be tested.

To avoid the program being too large for some VME CPU master board, which has not so much on board memory used for loading the program, we keep the two programs mentioned above separate.

PROGRAM 3 TMS32010 TRACE

This program is used to debug the TMS32010 application program. The hardware on FDDP board provides a possibility to generate a one micro second delayed interrupt signal when the instruction 'IN memory,port7' occurs.

It allows two instructions, enable interrupt and return from interrupt, to be executed during one micro second.

In the interrupt service routine of TMS32010, all the data in the internal registers are transfered to the out-FIFOs. Then the FIFOs are read by the master. And the data are displayed on the screen. The mailbox is used for communication between user and TMS. From one up to 32767 steps could be selected when tracing the user's application program.

The following instructions are needed to initialize the trace routine and must be loaded on the top of the user application program.

```
START   B       MAIN            Start address
        B       TRACE           Interrupt vector
        B       MAIN            Reserve 2 bytes
        B       MAIN            Reserve 2 bytes
MAIN    NOP                     Initialize
        LDPK    1               to 1
        LACK    1               step
        SACL    UNITY
        SACL    NUMSTP
        IN      TEMP,7          Set 1st interrupt
        EINT                    enable INT.
        NOP
        NOP
        NOP
USER    ......
```

The program is available by loading it, and loading the user's program in the VME CPU memory with offset $2000. The program starts at location $3000.All values in the internal registers will be dumped to the screen in the following format:

```
                 ***  TMS32010  TRACE Utility  ***

                       HEX                BINARY
ACCUMULATOR: xxxx.xxxx  xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
"P"REGISTER: xxxx.xxxx  xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
"T"REGISTER: xxxxxxxx.xxxxxxxx AUXREG0:xxxx AUXREG1:xxxxDP:x
STATUS REGISTER: xxxx  xxxxxxxx.xxxxxxxx  ARP:x OVM:x OV:x
PROGRAM COUNTER:xxxx.xxxxxxxx STACK0:xx STACK1:1:xxSTACK2:x
                   ***  Logical Unit Trace  ***
STATUS REGISTER: xxxxxxxx.xxxxxxxx CMD REGISTER: xxxx.xxxx
Tyte:^c=exit, N=change trace steps, anykey=trace, C=Continue
```

The TMS32010 Trace program provids an example of a cheaper and easier method of debuging Digital Signal Processor application programs, which is quite difficult without proper tools, such as a simulator.

ACKNOWLEDGEMENT

Shao Beibei has carried out this work with the suport of a grant from the International Centre for Theoretical Physics (ICTP) Program for Training and Research in Italian Laboratories, Trieste, Italy. He would like to express his thanks to Prof. A. Salam, the director of ICTP, Prof. G. Furlan, the head of the program, Prof. G. Denardo for providing him the opportunity to do research in the field of microprocessors. A special thanks to Prof. A.Marzari, Prof. G. Rinaudo and prof. F.Ferrero Director of the Istituto di Fisica Superiore dell' Univesita di Torino, for providing the hospitality. The authors gratefully acknowledges Prof. A. Werbrouck for suggestions.

REFERENCES

[1] D. Crosetto FDDP A Fast Digital Data Processor for the FEMC trigger DELPHI-87-70-DAS-58 July 1st 1987

[2] Horst von Eicken Software support for Motorola 68000 Microprocessor at CERN M68mil cross macro assembler CERN 83-12

[3] Lattice MC68000 Cross Macro Assembler and 'C' compiler for MS-DOS operating system

[4] QUELO MC68000 Cross Macro Assembler and 'C' compiler for MS-DOS operating system

[5] TEXAS INSTRUMENTS:TMS32010 User's Guide, 16/32-bit Digital Signal Processor Edited by Paul Strzelecki,B.Sc.