DIGITAL EQUIPMENT CORPORATION

COMPONENT COMPUTERS
ENGINEERING BULLETIN

COMPONENTS GROUP

JANUARY 1976

LSI 11 Engineering Bulletin

LSI11, PDP11/03 Differences

The purpose of this engineering bulletin is to document the specific differences existing between the several versions of the KD11-F and KD11-J processor modules, which serve as the CPU in LSI-11 and PDP11/03 microcomputer systems. Also included are programming recommendations pertinent to the several revisions.

At present, LSI-11 and PDP11/03 systems are equipped with processor modules having two levels of revision: one level is revisions C and D, and the other level is revision E and subsequent revisions. In the descriptions of differences and in programming recommendations, these two levels are referred to as Rev C/D and Rev E. The revision designation for a given processor module is indicated by the letter following the date code on the etch side of the processor module quick-release handle.

Each difference is defined in detail in this bulletin, along with necessary recommendations for resolving any departures from conventional PDP11 programming techniques imposed by the difference.

Revision Level Differences

Length of BUS INIT

The period of the signal BUS INIT, whether asserted by a RESET instruction or upon power up, will be approximately 100 μsec on Rev level C/D systems rather than the specified 12 μsec, as on the Rev level E systems. For those users having Rev level C/D systems, it is recommended that only the leading edge or BUS INIT be used and not the signal pulse width, when designing user interfaces.

RTT Instruction and Internal Refresh

If the T bit is set to 1 by an RTT instruction on systems at Rev level C/D, and the CPU performs a memory refresh immediately after executing the RTT, the one instruction delay for the T bit will not occur. As a result, the T bit trap will occur immediately after refresh is completed. The difference is present only when the CPU performs a refresh cycle so that the incidence should be very low since there is normally very little use of the T bit by user software. This difference can be eliminated on the present machines by having refresh done externally. As an alternative, this difference can be handled in a program using the T-bit, by making sure that the PC has changed from the previous trace trap. If the PC did not change, the program knows that a refresh cycle intervened.

On Rev level E systems, this one instruction cycle delay occurs normally after execution of an RTT.

Possibility of Memory Locations Being Altered When a BUS Error Occurs

Due to a race condition on Rev level C/D systems, the contents of locations 4, 6, 14, or 16 could possibly be altered during the processing of a bus error by the CPU, due to conflict on the WD tri-state DAL bus lines.

It is recommended that intentional bus errors, such as memory sizing, be kept to a minimum. This alteration, if it occurs, takes place before the content of the trap location is read by the CPU. Hence, if location 4 is altered, the resultant PC from that location would not be correct.

PSW Not Initialized for all Power-Up Options

If a power-up option other than to 24 (Option #0) is selected with the CPU module jumpers, on Rev level C/D systems, the event and device interrupts are inhibited until a specific MTPS instruction is executed. However, since the PSW bit 7 may be off, this condition is not apparent. Also, the remaining bits in the PSW are indeterminate at this time. On Rev C/D systems, for power-up modes other than to location 24, it is recommended that the PSW be initialized through the Stack Pointer (register R6). An MTPS instruction will not clear the T-bit, which could be set to one in this circumstance.

In the following example, a PC and the desired PSW are pushed onto the Stack, and an RTI is executed to load the PSW. The PC pushed onto the Stack is the address of the instruction following the RTI, so that after execution of the RTI, normal program execution can continue.

```
MOV #PSW   -(SP)   Push PSW onto Stack
MOV #TAG1-(SR)     Push new PC onto Stack
RTI                Pop new PC and PSW of
                   Stack
TAG: Program continues here
```

However, any trap or interrupt following power-up options #1, #2, or #3 should be avoided, because if the T bit is on, it will be saved on the stack and upon return, the T bit will be honored, since the micro-level flag is now set.

For Rev level E systems, the PSW is initialized to 200 so that interrupts are inhibited for the power-up to 173000 (option #2). The remaining power-up options, other than power-up to 24, specifically options #1 and #3, still have the above difference. If microcode or ODT power-up are selected (options #3 and 1 respectively), the PSW will not be initialized. For these two cases, the effect of this difference should be minimal since most users will type a "G" (GO) which will clear the PSW. However, those who select the ODT power-up option, then enter a program, set the PC, and type "P", could experience difficulty. To avoid this difficulty, the PSW should be explicitly cleared using the "RS" command.

Certain Reserved Op Codes in Group 7 through 77 Not Reserved

In the op code group 7 through 77, those codes that have bit 3=1 will be executed as the maintenance instruction (21R), rather than trapping to 10, by Rev level C/D systems. Those codes having bit 3=0 will try to execute microcode at micro PC 3000. But as long as a user does not add microcode that responds to micro PC 3000, these op codes will trap to 10.

Rubout Function Does Not Work for Register Numbers

This description clarifies a characteristic of Rev level C/D systems; specifically if a user types the following (where the underlined characters are those Micro-ODT characters previously typed):

@R5\6/012345	The response by the CPU is a backslash which is the symbol for RUBOUT.

This action will not open register 6(R6) but rather memory location 6. Once the "R" is typed, attempting to rubout only the register number will change modes and open the corresponding memory location instead. Other commands such as "line feed" will indicate that the user is in memory mode by typing memory locations and not register numbers. It is recommended that when the wrong register number is selected, another "R" be typed along with the new register number. For example, @R5R6/123456.

SRUN Differences

The signal SRUN is available at module finger CH1 on Rev E processor modules. (SRUN is described in the *LSI-11, PDP11/03 User's Manual,* Page 4-9.) This signal is asserted by Rev E processor modules under the following circumstances:

1. Once each time the CPU fetches an instruction.
2. Once each time the CPU executes a Reset instruction.
3. Twice each time the CPU responds to assertion of the bus signal BDCOK H.
4. Once each time the CPU times out on a bus cycle, (including timeouts occuring when in the terminal mode, and when sizing memory under the console "L" command).

In PDP11/03 systems, this signal is connected to a retriggerable one-shot to drive the front panel RUN indicator light.

On Rev C/D processor modules, SRUN does not appear on module finger CH1. However, on Rev C/D processor modules, SRUN can be decoded from the SROM fingers DD1, DE1, DF1, and DH1 (SROM signals SROM0 H, SROM1 H, SROM2 H and SROM3 H). These signals are stable and valid only while the signal SPH3 H on finger DD1 is true, so that SRUN can be decoded once each time the CPU fetches an instruction. Decoding of the SROM signals is based on SROM1 being true and SROM0, SROM2, and SROM3 being false, with SPH3 H

true serving as the decoding strobe. On Rev C/D processor modules this particular assertion of the SROM signals occurs once during the fetching of each instruction and at no other time.

Condition Codes Not Restored to Original Values After EIS/FIS Interrupt Abort

This difference applies only to diagnostics that may check this condition. Otherwise, it is transparent to a programmer since all the EIS and FIS instructions affect all the condition codes.

Programming Recommendations

Micro-ODT Typeout on Power-Up is Dependent on the Character in the UAR/T Buffer (DLV-11)

For both Rev level C/D and E systems and jumpered for power-up to location 24 option, the normal micro-ODT typeout is the contents of the PC followed by a carriage return, line feed, and prompt character. If that character is a RUBOUT (177), the typeout will be the contents of the PC followed by a backslash as a result of a RUBOUT character in the UAR/T buffer. The content of the UAR/T buffer can be a RUBOUT character as a consequence of DLV-11 power up. The carriage return, line feed, and prompt character will not be typed in this case. However, the state of micro-ODT does not change and after the backslash it is ready to accept commands. Consider this example: A power-up to 24 (24 contains 0 and 0 contains 0) normally would type:

```
000002
@
```

If a RUBOUT character is contained in the UAR/T buffer, the typeout would be

```
00002/
```

Although no prompt character is typed, ODT will accept all commands in the normal manner. Also, if the machine halts for any reason and a RUBOUT character is in the UAR/T input buffer, the same erroneous typeout will occur.

Caution on Storing MTPS and EIS Instructions in ROM

If MTPS and EIS instructions are executed out of ROM on both Rev level C/D and E systems, some addressing modes will cause a bus timeout error. This occurs because the processor performs a read-modify-write rather than a read when obtaining the source operand. Therefore, any mode where the effective source operand address is stored in ROM will cause a timeout error.

The following example causes a timeout:

```
MUL #123,R0
```

The timeout can be avoided, however, by first moving the literal to a general register or to RAM, as follows:

```
MOV #123,R4
MUL R4,R0
```

or another alternative is:

```
MOV #123,TEMP
MUL TEMP,R0
```

where TEMP is in RAM.

Event Line not Disabled when "G" and "L" Commands are Used

If a free-running clock, such as 60 Hz from the power supply, is attached to the BEVNT bus line on both Rev level C/D and E systems, an interrupt to location 100 will occur when using the "G" and "L" commands prior to executing the first instruction. Therefore, with an MTPS as the first instruction, a program can not disable the BEVNT bus line by inhibiting interrupts.

User programs requiring a free-running clock attached to the BEVNT bus line can temporarily avoid this situation by setting the PSW (RS) to 200, loading the PC with the starting address instead of using the "G" command, and then using the "P" command. Before using the "L" command, the PSW (RS) can be set to 200, thereby inhibiting interrupts, to avoid receiving the event interrupt after loading the ABS loader.

T Bit is On and a WAIT Instruction is Executed

If the T bit is = 1 when a WAIT instruction is executed on both Rev level C/D and E systems, the processor will hang and will not honor any device, event, halt, or power-down interrupts. However, refresh will still occur. In general, this situation will occur when an application program is linked to ODT-11, and the user is single stepping his program and a WAIT instruction is encountered. To escape this state, the processor must be powered-down and then powered-up again, with an assertion of the DCOK signal. Note that the content of dynamic MOS RAM could be lost if DCOK is asserted for longer than 400 μs. The use of a one-shot multivibrator to assert DCOK is recommended in these circumstances. This difference will not effect user software since the T bit is not generally used by application programs.

T Bit and EIS/FIS Instructions Being Aborted by Device Interrrupts

If the T bit is set while executing an EIS or FIS instruction, a subsequent event interrupt causes that instruction to be aborted, and the new PSW at the T bit vector location plus two (location 16) inhibits interrupts. Consequently, the processor will enter an endless loop executing the EIS or FIS instruction and the T bit trap and will never service the device or event interrupt.

The reason for this circumstance is that the RTT instruction, which is at the end of the T bit service routine, does not arbitrate interrupts if the T bit has been set, thereby popping the old PSW from the stack.

This restriction is imposed in order to guarantee that the T-bit trap is not immediately executed in this situation as with the RTI instruction. Most system and application software do not use the T bit and only a utility such as ODT-11 would use this facility. In those cases where the T bit is used, the solution is to allow interrupts to be honored in the T bit service routine by lowering the PS priority. Irrespective of T bit state, the limiting case concerns an interrupt such as a free-running clock tied to the BEVNT bus line which is occurring at such a rapid rate that the EIS or FIS instruction never finishes. In such a case, every time the interrupt is serviced and a return is made to re-try the EIS or FIS instruction, the interrupt occurs again and the instruction is aborted.

Caution When Clearing Device Interrupt Enable Bits

On both Rev level C/D and E systems, clearing device Interrupt Enable bits while the device is still active can lead to a bus time-out error when the processor attempts to receive the interrupt vector from that device. Consider the example:

```
PSW = 0
CLR @ #177564
```

As a result, the DLV-11 Serial Line Unit interrupt enable bit is being cleared. Now, assume that the transmitter is still active and sending characters, and further assume that the Done bit in the status register becomes set shortly after the CLR instruction is fetched, but before the Interrupt Enable bit can be cleared. The device will now post an interrupt request because Done bit has been set and Interrupt Enable bit is still set. The CLR instruction will complete execution and the processor will recognize the interrupt request since there was not enough time for the device to disable the interrupt request. The processor will then attempt to obtain a vector from the interrupting device. However, a bus time-out error will occur because the device now has had enough time to remove the interrupt request and will not respond. The processor treats this time-out as a fatal condition and halts by entering Micro-ODT. If multiple interrupt requests were pending at this time, a time-out would not occur since the next device would respond with an interrupt vector.

One method of avoiding this problem is to disable interrupts immediately before the Interrupt Enable bit is cleared. For example:

```
MTPS #200
CLR @#177564
MTPS #0
```

In this situation, enough time has been allowed for the interrupt request to be removed by the device. This feature was included to permit detection of faulty interrupt operation; specifically when an interrupting device does not properly respond within the required time period.

digital
DIGITAL EQUIPMENT CORPORATION
COMPONENTS GROUP HEADQUARTERS
ONE IRON WAY, MARLBOROUGH, MASSACHUSETTS 01752
(617) 481-7400 TWX: 710-347-0348
For detailed information about products and policies, call 800-225-9480 toll-free (USA only); Massachusetts residents call (617) 481-7400.