

DESIGNER'S NOTEBOOK



Accessing Status and Control Fields and I/O Ports in the TMS320Cxx HLL Debugger

Contributed by Jay Reimer

Design Problem

How can I observe the individual fields in the status registers in a convenient way? As an example, I want to observe the present or past auxiliary register pointer (ARP) of the TMS320C5x when I stop at a breakpoint or single step.

How can I read and write the I/O ports on the 'C5x while using the HLL debugger without writing 'C5x assembly code?

Solution

The ARP along with the data page pointer (DP), carry bit (C), and a variety of other fields are located in the status registers of the 'C5x. These registers are displayed in the CPU register window of the HLL debugger but it is inconvenient to observe the value of any one individual field.

The HLL debugger has two features that can help display the desired field(s) in an easy-to-observe manner. These two features are the ALIAS command and the WATCH window.

The ALIAS command and the WATCH window are also useful in reading and writing I/O ports without having to write 'C5x assembly code. This can be especially useful during debug when you want to observe an input port value at some arbitrary point in your code or you need to write a new value to an output port.

The examples in this report provide a list of alias commands that can be useful in displaying the fields of interest and alias commands that are useful in manipulating I/O ports. These examples are specific to the 'C5x, but can easily be adapted for any of the other generations of TMS320 DSPs.

Since the debugger does not support access of status register fields natively, it requires writing an expression. To avoid having to rewrite these expressions with each debug session, the aliases were written and stored in a file "regs.cmd". If you add

```
take regs.cmd,0
```

to your evminit.cmd file, these aliases will be loaded each time you invoke the debugger. (The ,0 suppresses echo of the aliases in the command window during load.)

You use these alias commands in the following ways:

```
wa-DP                ;add watch for DP
wa-ARP               ;add watch for ARP

?-DP                ;echo DP value in command window
?-XF                ;echo XF value in command window

e-ARP 4              ;change ARP value to 4
e-TC 1               ;change TC value to 1
```

```
; Status Register 0 (ST0) Fields
alias wa-DP,         "wa ST0&0x1ff,DP"
alias wa-INTM,       "wa (ST0>>9)&1,INTM"
alias wa-OVM,        "wa (ST0>>11)&1,OVM"
alias wa-OV,         "wa (ST0>>12)&1,OV"
alias wa-ARP,        "wa (ST0>>13)&7,ARP"

alias ?-DP,          "? ST0&0x1ff"
alias ?-INTM,        "? (ST0>>9)&1"
alias ?-OVM,         "? (ST0>>11)&1"
alias ?-OV,          "? (ST0>>12)&1"
alias ?-ARP,         "? (ST0>>13)&7"

alias e-DP,          "e ST0 = (ST0&0xfe00)+(%1 & 0x1ff)"
alias e-INTM,        "e ST0 = (ST0&0xfdff)+((%1 & 1)<<9)"
alias e-OVM,         "e ST0 = (ST0&0xf7ff)+((%1 & 1)<<11)"
alias e-OV,          "e ST0 = (ST0&0xffff)+((%1 & 1)<<12)"
alias e-ARP,         "e ST0 = (ST0&0x1fff)+((%1 & 7)<<13)"

; Status Register 1 (ST1) Fields
alias wa-PM,         "wa ST1&3,PM"
alias wa-XF,         "wa (ST1>>4)&1,XF"
alias wa-HM,         "wa (ST1>>6)&1,HM"
alias wa-C,          "wa (ST1>>9)&1,C"
alias wa-SXM,        "wa (ST1>>10)&1,SXM"
alias wa-TC,         "wa (ST1>>11)&1,TC"
alias wa-CNF,        "wa (ST1>>12)&1,CNF"
alias wa-ARB,        "wa (ST1>>13)&7,ARB"

alias ?-PM,          "? ST1&3"
alias ?-XF,          "? (ST1>>4)&1"
alias ?-HM,          "? (ST1>>6)&1"
alias ?-C,           "? (ST1>>9)&1"
alias ?-SXM,         "? (ST1>>10)&1"
alias ?-TC,          "? (ST1>>11)&1"
alias ?-CNF,         "? (ST1>>12)&1"
alias ?-ARB,         "? (ST1>>13)&7"

alias e-PM,          "e ST1 = (ST1&0xffffc) + %1 & 3"
alias e-XF,          "e ST1 = (ST1&0xffef) + ((%1 & 1)<<4)"
alias e-HM,          "e ST1 = (ST1&0xffbf) + ((%1 & 1)<<6)"
alias e-C,           "e ST1 = (ST1&0xfdff) + ((%1 & 1)<<9)"
alias e-SXM,         "e ST1 = (ST1&0xfbff) + ((%1 & 1)<<10)"
alias e-TC,          "e ST1 = (ST1&0xf7ff) + ((%1 & 1)<<11)"
alias e-CNF,         "e ST1 = (ST1&0xffff) + ((%1 & 1)<<12)"
alias e-ARB,         "e ST1 = (ST1&0x1fff) + ((%1 & 7)<<13)"
```

Figure 1: regs.cmd file listing

```

; Processor Mode Status Register (PMST) Fields
alias wa-BRAF,    "wa PMST&1,BRAF"
alias wa-TRM,    "wa (PMST>>1)&1,TRM"
alias wa-NDX,    "wa (PMST>>2)&1,NDX"
alias wa-MPMC,   "wa (PMST>>3)&1,MPMC"
alias wa-RAM,    "wa (PMST>>4)&1,RAM"
alias wa-OVLY,   "wa (PMST>>5)&1,OVLY"
alias wa-AVIS,   "wa (PMST>>7)&1,AVIS"
alias wa-IPTR,   "wa (PMST>>11)&0x1f,IPTR"

alias ?-BRAF,    "? PMST&1"
alias ?-TRM,    "? (PMST>>1)&1"
alias ?-NDX,    "? (PMST>>2)&1"
alias ?-MPMC,   "? (PMST>>3)&1"
alias ?-RAM,    "? (PMST>>4)&1"
alias ?-OVLY,   "? (PMST>>5)&1"
alias ?-AVIS,   "? (PMST>>7)&1"
alias ?-IPTR,   "? (PMST>>11)&0x1f"

alias e-BRAF,    "e PMST=(PMST&0xfffe) + %1 & 1"
alias e-TRM,    "e PMST=(PMST&0xffffd) + ((%1 & 1)<<1)"
alias e-NDX,    "e PMST=(PMST&0xffffb) + ((%1 & 1)<<2)"
alias e-MPMC,   "e PMST=(PMST&0xffff7) + ((%1 & 1)<<3)"
alias e-RAM,    "e PMST=(PMST&0xffef) + ((%1 & 1)<<4)"
alias e-OVLY,   "e PMST=(PMST&0xffdf) + ((%1 & 1)<<5)"
alias e-AVIS,   "e PMST=(PMST&0xff7f) + ((%1 & 1)<<7)"
alias e-IPTR,   "e PMST=(PMST&0x07ff) + ((%1 & 0x1f)<<11)"

; Circular Buffer Control Register (CBCR) Fields
alias wa-CAR1,   "wa CBCR&7,CAR1"
alias wa-CENB1,  "wa (CBCR>3)&1,CENB1"
alias wa-CAR2,   "wa (CBCR>4)&7,CAR2"
alias wa-CENB2,  "wa (CBCR>7)&1,CENB2"

alias ?-CAR1,    "? CBCR&7"
alias ?-CENB1,   "? (CBCR>>3)&1"
alias ?-CAR2,    "? (CBCR>>4)&7"
alias ?-CENB2,   "? (CBCR>>7)&1"

alias e-CAR1,    "e CBCR=(CBCR&0xffff8) + %1 & 7"
alias e-CENB1,   "e CBCR=(CBCR&0xffff7) + ((%1 & 1)<<3)"
alias e-CAR2,    "e CBCR=(CBCR&0xff8f) + ((%1 & 7)<<4)"
alias e-CENB2,   "e CBCR=(CBCR&0xff7f) + ((%1 & 1)<<7)"

```

Figure 1: `regs.cmd` file listing (continued)

Since the debugger does not support access of I/O ports natively, it requires writing an expression. To avoid having to rewrite these expressions with each debug session I created aliases and stored them in a file `"io.cmd"`. If you add

```
take io.cmd,0
```

to your `evmunit.cmd` file these aliases will be loaded each time you invoke the debugger. (The `,0` suppresses echo of the aliases in the command window during load.)

You use these alias commands in the following ways:

```
IN 0          ; reads I/O port 0
IN 0xc        ; reads I/O port 0xc (12)

OUT 5,0xfe    ; writes 0xfe to I/O port 5
OUT 0xf,0     ; writes 0 to I/O port 0xf (15)
```

```
alias IN,      "? *%1@io"
alias OUT,     "e *%1@io = %2"
```

Figure 2: io.cmd file listing