TMS320 DSP
Number 15
DESIGNER'S NOTEBOOK
TEXAS INSTRUMENTS

Efficient Coding on the TMS320C5x

Contributed by Mansoor Chishtie

Design Problem What are some examples of software that take advantage of the 'C5x architecture?

Solution Algorithms based on dynamic programming techniques often make use of looped code, conditional execution, min-max search, and pointer addressing. The TMS320C5x core CPU allows zero-overhead looping, multiple-condition branches, delayed jumps and calls, min-max instructions, and post-modify indirect addressing to implement efficient search algorithms.

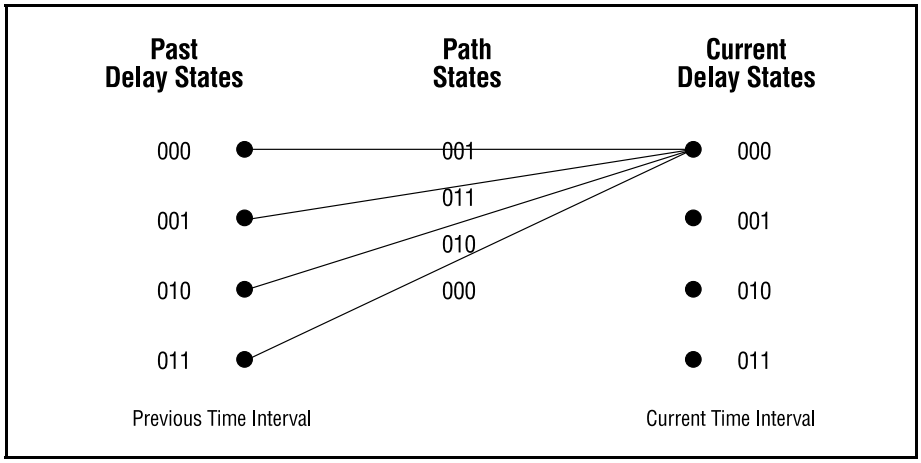

Figure 1. A popular Viterbi subroutine selects the "minimum cost" path

The Viterbi decoding algorithm is quite popular in data communications applications. One subroutine used by this algorithm is presented here to demonstrate efficient 'C5x code. The function of the subroutine is to select the "minimum cost" path to current delay state out of four possible paths (see Figure 1). Each path has its associated "cost value" and each past delay state has an accumulated cost. The new accumulated cost is computed by adding the path cost to the accumulated cost of the past delay state. The path with the minimum accumulated cost is selected and the rest are discarded.

Texas Instruments provides customer support in varied technical areas. Since TI does not possess full access to data concerning all of the uses and applications of customers' products, TI assumes no responsibility for customer product design or the use or application of customers' products or for any infringements of patents or rights of others which may result from TI assistance. 1/93

When a path link is selected, the path state value identifying the link and the past delay state are stored in appropriate tables (PAST_PTH, PAST_DLY). The acc. distance and current distance tables (ACCDIST, DIST) are accessed by indirect circular addressing mode. The four path states are not in binary ascending or descending order, but a four-word circular buffer can be set up that steps through each path state in the correct sequence. It also resets the pointer to the first state automatically after exiting the loop. Note that all the instructions inside the block-repeat loop are single-cycle instructions.

The complete 'C5x Viterbi implementation is available on the BBS.

* AR1 - ACC_DIST (set up as 4-word circular buffer)
* AR2 - DIST (set up as 4-word circular buffer)
* AR3 - MIN_DIST (minimum accumulated distance table)
* AR5 - PAST_DLY (past delay state table)
* AR6 - PAST_PTH (past path state table)

```
BEGIN
  MAR     *,AR1         ;ARP = AR1
  SPLK    #3,BRCR       ;set up count
  LACC    #07FFFH       ;max value
  SACB                  ;ACCB=07fffh

  RPTB    LOOP-1        ;repeat 4 times
  ACC     *,0,AR2       ;Get old acc distance
  ADD     *,0,AR3       ;Add current distance
  CRLT                     ;if (ACC ACCB)
                            then ACCB=ACC
  SACL    *,0,AR5       ;Save new min value
  XC      2,C           ;Update PAST_PTH
                         and PAST_DLY
  SAR     AR1,*,AR6     ;pointer to ACCDIST - PAST_DLY
  SAR     AR2,*,AR1     ;pointer to DIST - PAST_PTH
  MAR     *,AR1         ;ARP = AR1
  MAR     *+,AR2        ;AR1++ (circular addressing)
  MAR     *+,AR1        ;AR2++ (circular addressing)
LOOP
```