

# **TMS320DM642 Technical Overview**

---

*Vishal Markandey  
Dipa Rao*

*DSP Video and Imaging  
Digital Applications*

## **ABSTRACT**

The TMS320DM642 Digital Media Processor is based on the C64x CPU, which is part of the C6000 DSP family. The DM642 integrates a number of peripherals to address the development of video and imaging applications, including three configurable video ports capable of glueless video input, video output, or transport stream input. These video ports can support BT.656 video I/O, HDTV Y/C I/O at upto 10-bits per component, RGB I/O, MPEG-2 Transport stream inputs. Also included on the DM642 is a 10/100 Mbps ethernet MAC (EMAC), a multichannel audio serial port (McASP) that supports common audio interface modes including I2S and DIT, a 66 MHz 32-bit PCI bus, and a number of other peripherals.

This document describes the DM642 architecture including details of its peripherals. Several example applications are described, showing how DM642 can be used in development of video IP phones, video-on-demand set-top boxes, and surveillance digital video recorders.

---

## **Contents**

<b>Introduction</b> .....	<b>3</b>
<b>DM642 CPU</b> .....	<b>5</b>
Register Files .....	6
Functional Units .....	6
Register File Paths .....	8
Memory, Load and Store Paths .....	9
Packed Data Processing .....	10
Additional Functional Unit Hardware .....	11
<b>DM642 Cache Architecture</b> .....	<b>12</b>
<b>DM642 Enhanced DMA (EDMA) Controller</b> .....	<b>13</b>
<b>DM642 Video Port</b> .....	<b>13</b>
Video Port Architecture .....	14
Video Port Signals .....	15
Video Capture Operation .....	16
BT.656 Video Capture Mode .....	16
Y/C Video Capture Mode .....	17
Ancillary Data Capture .....	18
Raw Video Capture Modes .....	19

Trademarks are the property of their respective owners.

TSI Capture Mode .....	19
Video Input Filtering .....	20
Chrominance Resampling .....	20
1/2 Scaling .....	21
Video Display Operation .....	22
BT.656 Video Display Mode .....	22
Y/C Video Display Mode .....	23
Ancillary Data Display .....	24
Raw Video Display Mode .....	24
Video Output Filtering .....	25
Chrominance Resampling Operation .....	25
2x Scaling Operation .....	25
<b>VCXO Interpolated Control Port (VIC) .....</b>	<b>27</b>
<b>Ethernet Media Access Controller (EMAC) .....</b>	<b>28</b>
Interface Feature Set .....	28
EMAC Operation .....	28
<b>Management Data Input/Output (MDIO) .....</b>	<b>29</b>
<b>Multichannel Audio Serial Port (McASP) .....</b>	<b>30</b>
<b>Other DM642 Peripherals .....</b>	<b>30</b>
External Memory Interface (EMIF) .....	30
Host Port Interface (HPI) .....	31
Peripheral Component Interconnect (PCI) .....	31
Multichannel Buffered Serial Ports (McBSPs) .....	31
General Purpose I/O Support .....	31
Additional Peripheral Information .....	32
<b>DM642 Peripherals Multiplexing .....</b>	<b>32</b>
<b>DM642 Application Examples .....</b>	<b>33</b>
Video IP Phone .....	34
Video on Demand Set-Top Box .....	36
Surveillance Digital Video Recorder .....	38
<b>Ease of Development .....</b>	<b>39</b>
<b>Summary .....</b>	<b>40</b>
<b>Appendix A Comparison of DM642 ot C64x DSPs .....</b>	<b>41</b>
<b>Appendix B Related Documents .....</b>	<b>42</b>

### List of Figures

Figure 1. DM642 Block Diagram .....	4
Figure 2. CPU for VelociTI.2 .....	6
Figure 3. C64x Data Cross Paths .....	9
Figure 4. C64x Memory Load and Store Paths .....	9
Figure 5. DM642 L1/L2 Cache .....	12
Figure 6. Video Port Block Diagram .....	15
Figure 7. BT.656 Video Capture .....	17
Figure 8. Y/C Video Capture .....	18

Figure 9. MPEG-2 Transport Stream Capture .....	20
Figure 10. Video Input Chrominance Resampling .....	21
Figure 11. Video Input 1/2 Scaling .....	21
Figure 12. Video Input 1/2 Scaling With Chrominance Resampling .....	22
Figure 13. BT.656 Display .....	23
Figure 14. RGB Display .....	24
Figure 15. Video Output Chrominance Resampling .....	25
Figure 16. Video Output 2x Scaling .....	26
Figure 17. Video Output 2x Scaling With Chrominance Resampling .....	26
Figure 18. PCR Header Format .....	27
Figure 19. DM642 Networking Using EMAC and MDIO .....	29
Figure 20. DM642 Peripherals Multiplexing .....	33
Figure 21. DM642 Based Video IP Phone .....	35
Figure 22. Set-Top Box Application Using a DM642 .....	37
Figure 23. Networked DVR System Using a DM642 .....	39

### List of Tables

Table 1. Quad 8-Bit and Dual 16-Bit Instruction Set Extensions .....	11
Table 2. C64x Special Purpose Instructions for Video and Imaging .....	11
Table 3. Video Capture Modes .....	16
Table 4. Video Display Modes .....	22
Table A-1. Comparison of DM642 to C64x DSPs .....	41

## Introduction

Programmable digital signal processors (DSPs) are increasingly important in a wide range of video and imaging applications, such as machine vision, medical imaging, security monitoring, digital cameras and printers, and a large number of consumer applications driven by digital video processing including DVDs, digital TV, video telephony, and many others. These applications are characterized by requirements for processing flexibility, sophisticated algorithms, and high data rates.

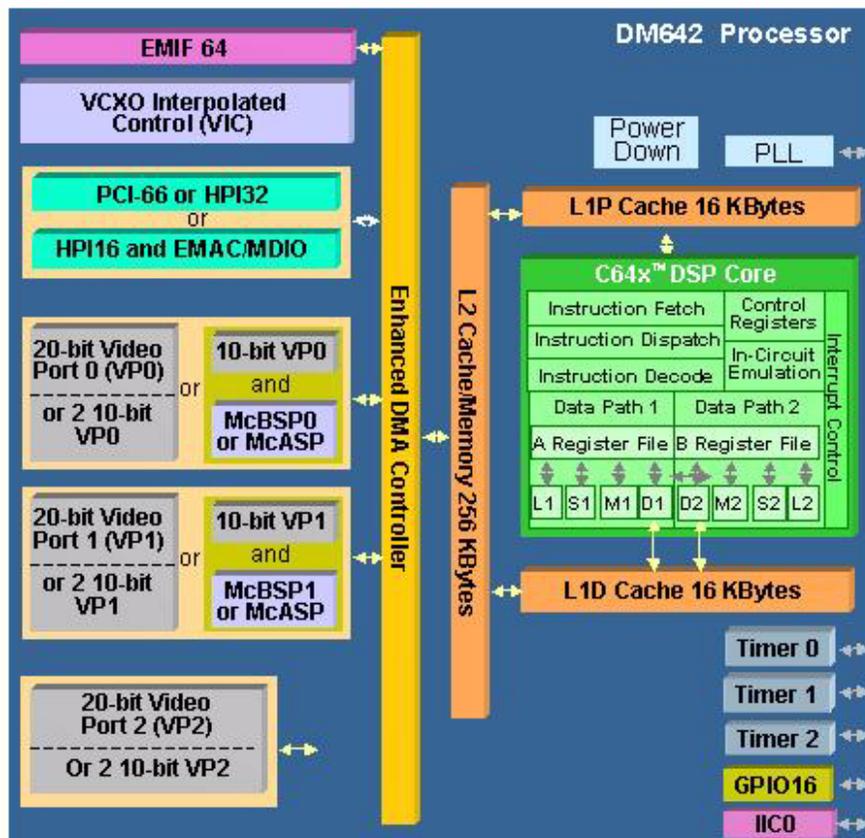
The C6000 DSP family with the VelociTI architecture addresses the needs of video and imaging applications. First introduced in 1997 with the C62x and C67x cores, the C6000 family uses an advanced very long instruction word (VLIW) architecture. The architecture contains multiple execution units running in parallel, which allow them to execute multiple instructions in a single clock cycle. Parallelism is the key to extremely high performance. At a 200 MHz clock rate and 1600 million instructions per second (MIPS) at introduction, the C6201 achieved ten times the performance of earlier digital signal processing solutions.

In 2001 TI achieved another significant performance increase with the introduction of the C64x. Initial devices sampled in the 500 MHz-600 MHz range gave performance levels of 4000-4800 MIPS. In addition to clock rate, C64x introduced VelociTI.2 extensions to the VelociTI architecture. These extensions allow more work to be done in each cycle by including new instructions to accelerate performance in key application areas including video and imaging.

Increased clock rate and increased CPU throughput are only part of the solution. Processing data at extremely high rates increases the need for I/O bandwidth. Initial C64x devices have three external buses with speeds of up to 133 MHz. Each bus has a primary mission. One provides a fast glueless interface for synchronous and asynchronous memories at data rates as fast as 1.1Gbytes/sec. Another bus interfaces to slow peripheral devices and the third bus provides a port to support industry standard host interfaces. Flexible multichannel buffered serial ports can each supply 100Mbps/sec of additional throughput. The internal DMA engine can provide over 2Gbytes/sec of I/O bandwidth with 64 independent channels.

Developers of video and imaging applications require peripherals specific to video and imaging, for system integration support. Examples include glueless interfaces for BT.656 video I/O, HDTV Y/C I/O at upto 10-bits per component, RGB I/O, MPEG-2 transport stream interfaces, ethernet MAC support, and audio interfaces for I2S, DIT etc. The DM642 provides a rich suite of on-chip peripherals that address these system interface requirements.

The DM642, shown in Figure 1-1, builds upon the C64x core and peripheral set by introducing the following new peripherals:



**Figure 1. DM642 Block Diagram**

- Three configurable video ports capable of glueless video Input, video output, or transport stream input
- VCXO interpolated control port (VIC)
- 10/100 Mbps ethernet MAC (EMAC)

- Management data input/output (MDIO) module
- Multichannel audio serial port (McASP)
- Inter-integrated circuit (I2C) bus module
- Two multichannel buffered serial ports (McBSPs)
- Three 32-bit general-purpose timers
- User-configurable 16-bit or 32-bit host port interface (HPI16/HPI32)
- 66 MHz 32-bit peripheral component interconnect (PCI)
- General-purpose input/output (GPIO) port
- 64-bit external memory interface (EMIF) that is capable of interfacing to synchronous and asynchronous memories and peripherals

## DM642 CPU

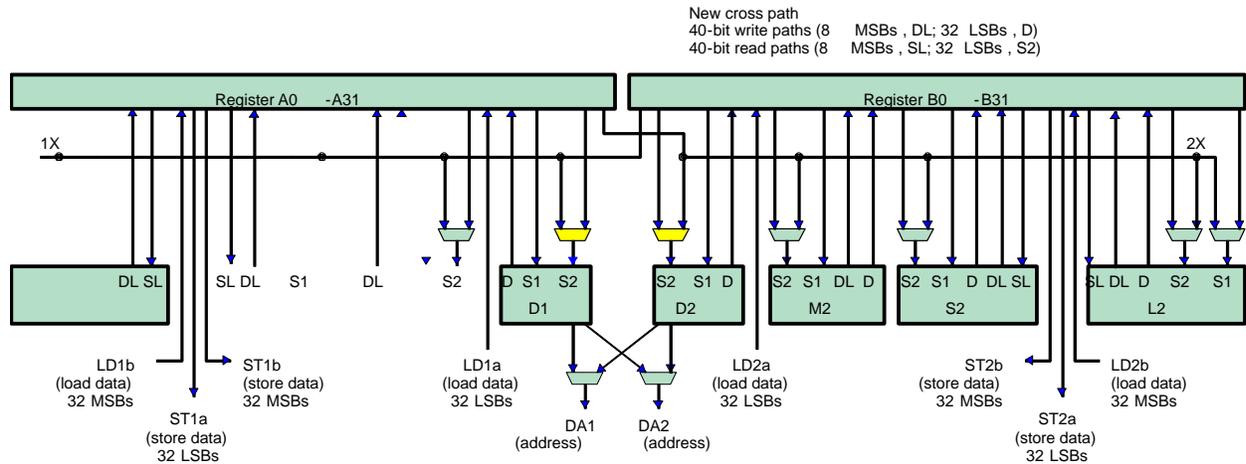
The DM642 is based on the C64x CPU, which is part of the C6000 DSP family. The C6000 DSP family with the VelociTI architecture addresses the needs of video and imaging applications. The C6000 family uses an advanced very long instruction word (VLIW) architecture. The architecture contains multiple execution units running in parallel, which allow them to execute multiple instructions in a single clock cycle. Parallelism is the key to extremely high performance. The C64x adds another significant performance boost to the C6000 DSP family. In addition to clock rate, C64x introduced VelociTI.2 extensions to the VelociTI architecture. These extensions allow more work to be done in each cycle by including new instructions to accelerate performance in key application areas including video and imaging.

The C6000 CPU components consist of:

- Two general-purpose register files (A and B)
- Eight functional units (.L1, .L2, .S1, .S2, .M1, .M2, .D1, and .D2)
- Two load-from-memory data paths (LD1 and LD2)
- Two store-to-memory data paths (ST1 and ST2)
- Two data address paths (DA1 and DA2)
- Two register file data cross paths (1X and 2X)

Figure 2 illustrates the VelociTI.2 CPU.

C64x CPU - VelociTI .2



**Figure 2. CPU for VelociTI.2**

## Register Files

There are two general-purpose register files (A and B) in the C6000 data paths. For the C64x, each of these files contains 32 32-bit registers (A0–A31 for file A and B0–B31 for file B). The general-purpose registers can be used for data, data address pointers, or condition registers. On the C64x, registers A0, A1, A2, B0, B1, and B2 can be used as condition registers. In all C6000 devices, registers A4–A7 and B4–B7 can be used for circular addressing.

The C64x register file supports data ranging in size from packed 8-bit data, packed 16-bit data, through 40-bit fixed-point, 64-bit fixed point, and 64-bit floating-point data. Values larger than 32 bits, such as 40-bit long and 64-bit float quantities, are stored in register pairs, with the 32 LSBs of data placed in an even-numbered register and the remaining 8 or 32 MSBs in the next upper register (which is always an odd-numbered register). Packed data types store either four 8-bit values or two 16-bit values in a single 32-bit register or four 16-bit values in a 64-bit register pair.

## Functional Units

The eight functional units in the C6000 data paths can be divided into two groups of four; each functional unit in one data path is almost identical to the corresponding unit in the other data path. The C64x contains many 8-bit and 16-bit instructions to support video and imaging applications. The C64x functional units and their capabilities are listed below:

- |                    |                                                  |
|--------------------|--------------------------------------------------|
| .M unit (.M1, .M2) | 16x16 multiply operations                        |
|                    | 16x32 multiply operations                        |
|                    | Quad 8x8 multiply operations                     |
|                    | Dual 16x16 multiply operations                   |
|                    | Dual 16x16 multiply with add/subtract operations |
|                    | Quad 8x8 multiply with add operations            |
|                    | Bit expansion                                    |

	Bit interleaving/de-interleaving
	Galois field multiply
	Rotation
	Variable shift operations
.L unit (.L1, .L2)	32/40-bit arithmetic and compare operations
	32-bit logical operations
	Leftmost 1 or 0 counting for 32-bits
	Normalization count for 32-bits and 40-bits
	Byte shifts
	Data packing/unpacking
	5-bit constant generation
	Dual 16-bit arithmetic operations
	Quad 8-bit arithmetic operations
	Dual 16-bit min/max operations
	Quad 8-bit min/max operations
	Quad 8-bit subtract with absolute value
.S unit (.S1, .S2)	32-bit arithmetic operations
	32-bit/40-bit shifts and 32-bit bit-field operations
	32-bit logical operations
	Branches
	Constant generation
	Register transfers to/from control register file (.S2 only)
	Byte shifts
	Data packing/unpacking
	Dual 16-bit compare operations
	Quad 8-bit compare operations
	Dual 16-bit shift operations
	Dual 16-bit saturated arithmetic operations
	Quad 8-bit saturated arithmetic operations
.D unit (.D1, .D2)	32-bit add, subtract, linear and circular address calculation

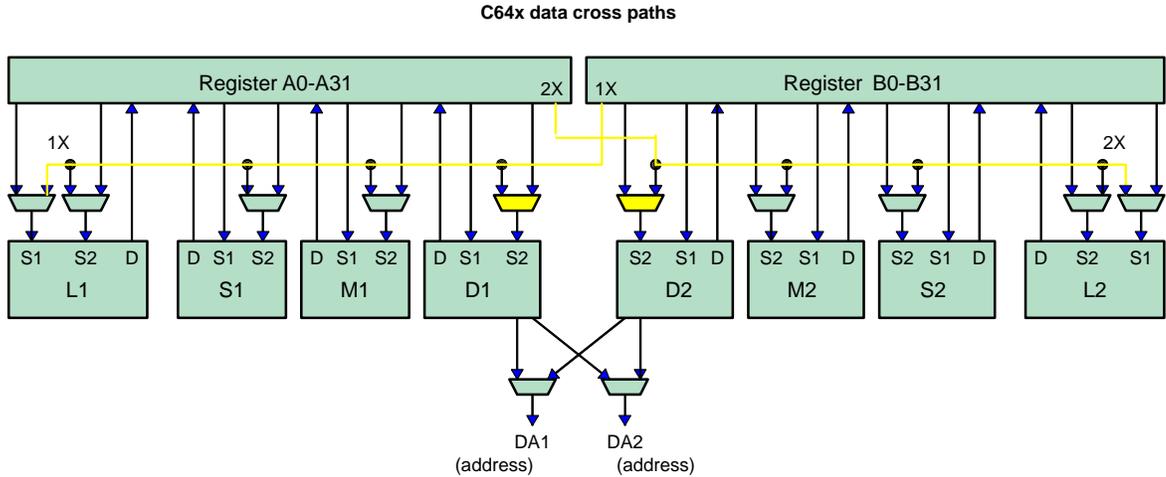
- Loads and stores with 5-bit constant offset
- Loads and stores with 15-bit constant offset (.D2 only)
- Load and store double words with 5-bit constant offset
- Load and store non-aligned words and double words
- 5-bit constant offset generation
- 32-bit logical operations
- Dual 16-bit arithmetic operations

## Register File Paths

Each functional unit reads directly from and writes directly to the register file within its own data path. That is, the .L1, .S1, .D1, and .M1 units write to register file A, and the .L2, .S2, .D2, and .M2 units write to register file B.

Most data lines in the CPU support 32-bit operands, and some support long (40-bit) and double word (64-bit) operands. Each functional unit has its own 32-bit write port into a general-purpose register file (Figure 2). Each functional unit has two 32-bit read ports for source operands *src1* and *src2*. Four units (.L1, .L2, .S1, and .S2) have an extra 8-bit-wide port for 40-bit long writes, as well as an 8-bit input for 40-bit long reads. Because each unit has its own 32-bit write port, all eight units can be used in parallel with every cycle when performing 32 bit operations. Since each C64x multiplier can return up to a 64-bit result, an extra write port has been added from the multipliers to the register file, as compared to the C62x.

The register files are also connected to the opposite-side register file's functional units via the 1X and 2X cross paths (Figure 3). These cross paths allow functional units from one data path to access a 32-bit operand from the opposite side's register file. The 1X cross path allows functional units from data path A to read its source from register file B. Similarly, the 2X cross path allows functional units from data path B to read its source from register file A.



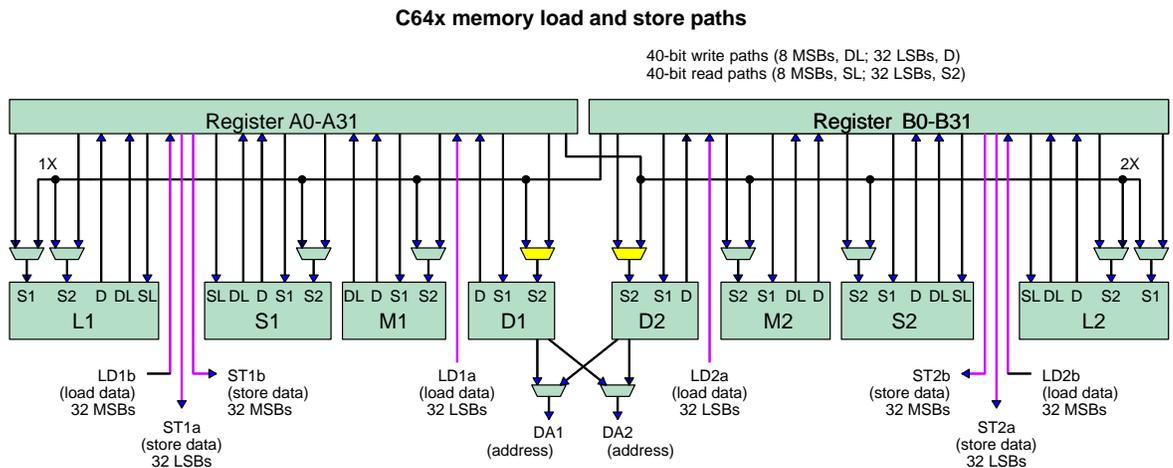
**Figure 3. C64x Data Cross Paths**

On the C64x, all eight of the functional units have access to the register file on the opposite side via a cross path. The .M1, .M2, .S1, .S2, .D1 and .D2 units' *src2* inputs are selectable between the cross path and the register file found on the same side. In the case of the .L1 and .L2, both *src1* and *src2* inputs are also selectable between the cross path and the same-side register file.

Only two cross paths, 1X and 2X, exist in the C6000 architecture. Therefore, the limit is one source read from each data path's opposite register file per cycle, or a total of two cross-path source reads per cycle. The C64x pipelines data cross path accesses allow multiple units per side to read the same cross-path source simultaneously. The cross path operand for one side may be used by up to two functional units on that side in an execute packet.

### Memory, Load and Store Paths

The data address paths named DA1 and DA2 are each connected to the .D units in both data paths. Load/store instructions can use an address register from one register file while loading to or storing from the other register file. Figure 4 illustrates the C64x memory load and store paths.



**Figure 4. C64x Memory Load and Store Paths**

The C64x device supports double-word loads and stores. There are four 32-bit paths for loading data for memory to the register file. For side A, LD1a is the load path for the 32 LSBs; LD1b is the load path for the 32 MSBs. For side B, LD2a is the load path for the 32 LSBs; LD2b is the load path for the 32 MSBs. There are also four 32-bit paths for storing register values to memory from each register file. ST1a is the write path for the 32 LSBs on side A; ST1b is the write path for the 32 MSBs for side A. For side B, ST2a is the write path for the 32 LSBs and ST2b is the write path for the 32 MSBs. Wide loads are essential in sustaining processing throughput.

The C64x device can also access words and double words at any byte boundary using non-aligned loads and stores. As a result, word and double-word data does not always need alignment to 32-bit or 64-bit boundaries. This feature is particularly useful in motion estimation and video filtering operations, where one may need access to data from any arbitrary byte boundary in memory. Non-aligned loads and stores combined with the pack and unpack instructions described earlier, mean that the compiler does not have to format the data to take advantage of the 8-bit and 16-bit hardware extensions. Without these operations, significant effort would be needed to leverage the parallelism. C64x provides a complete set of data flow operations to sustain the maximum performance improvement made possible by the 8-bit and 16-bit extensions added to the C6000 architecture.

## Packed Data Processing

C64x includes instruction that operate directly on packed data (both 8-bit and 16-bit) to streamline data flow and increase instruction set efficiency. An extensive collection of pack and unpack instructions simplifies manipulation of packed data types. The C64x has a comprehensive collection of 8-bit and 16-bit instruction set extensions. They are included in Table 1.

**Table 1. Quad 8-Bit and Dual 16-Bit Instruction Set Extensions**

Operation	Quad 8-Bit	Dual 16-Bit
Multiply	X	X
Multiply with saturation		X
Addition/subtraction	X	X†
Addition with saturation	X	X
Absolute		X
Subtract with absolute value	X	
Compare	X	X
Shift		X
Data pack/unpack	X	X
Data pack with saturation	X	X
Dot product with optional negate	X‡	X
Min/max/average	X	X
Bit-expansion (mask generation)	X	X

† The C62x/C67x provides support for 16-bit data with the ADD2/SUB2 instructions. The C64x extends this support to include 8-bit data.

‡ Dot product with negate is not available for 8-bit data

## Additional Functional Unit Hardware

Additional hardware has been built into the eight functional units of the C64x. We have already discussed two important extensions. Each .M unit can perform two 16x16 bit multiplies or four 8x8 bit multiplies every clock cycle. Also, the .D units can access words and double words on any byte boundary by using non-aligned load and store instructions.

In addition, the .L units can perform byte shifts and the .M units can perform bi-directional variable shifts in addition to the .S unit's ability to do shifts. The .L units can perform quad 8-bit subtracts with absolute value. This absolute difference instruction greatly aids motion estimation algorithms. Table 2 lists some of the special purpose instructions included in C64x for video and imaging applications.

**Table 2. C64x Special Purpose Instructions for Video and Imaging**

Instruction	Description	Example Application
BITC4	Bit count	Machine vision
LDNDW	Load non-aligned double word	Motion estimation, image filtering
DOTPx	Dot product	Image filtering, image resizing, image transforms (e.g. DCT, wavelet)
MINUx/MAXUx	Minimum/maximum computation	Nonlinear image filtering (e.g. median filter)
PACKx	Data packing and unpacking	Multiplexing/demultiplexing interleaved data (e.g. Y/C data)
XPNDx	Bit expansion	Graphics

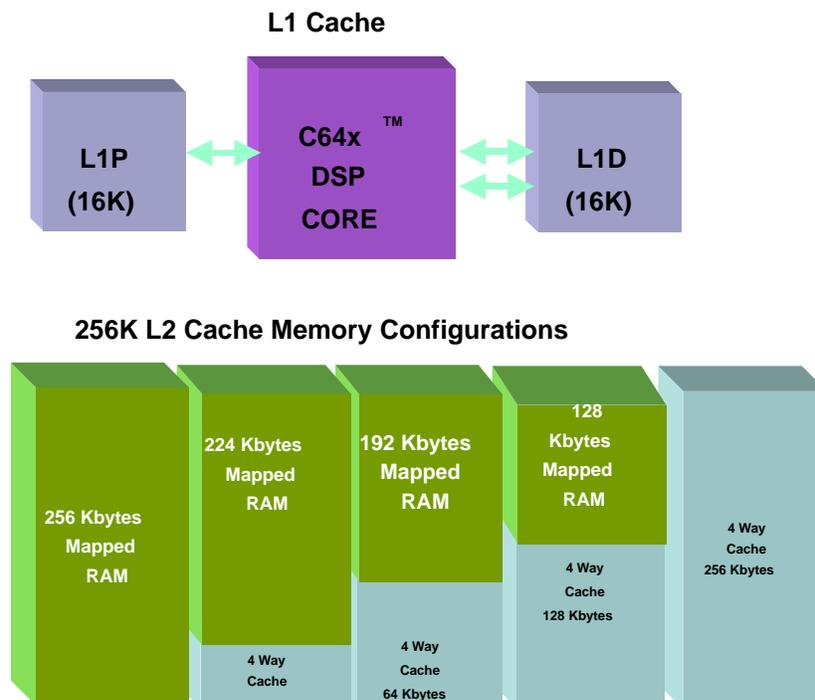
**Table 2. C64x Special Purpose Instructions for Video and Imaging (Continue)**

Instruction	Description	Example Application
CMPx	Compare	Image thresholding
AVGx	Quad 8-Bit, dual 16-bit average	Motion compensation
SUBABS4	Quad 8-bit absolute of differences	Motion estimation
SHFL/DEAL	Bit interleaving/deinterleaving	Bit planarization

It is important to note that the C64x provides a comprehensive set of data packing and unpacking operations to allow sustained high performance for the quad 8-bit and dual 16-bit hardware extensions. Unpack instructions prepare 8-bit data for parallel 16-bit operations. Pack instructions return parallel results to output precision including saturation support.

## DM642 Cache Architecture

On DM642 devices, the CPU interfaces directly to dedicated level-one program (L1P) and data (L1D) caches of 16 Kbytes each. These caches operate at the full speed of CPU access. A second level unified L2 program/data memory provides flexible storage. Figure 5 depicts an example L2 of size 256 Kbytes; the size and segmentation of the L2 cache in the DM64x family may change over time. One configuration for L2 is entirely mapped SRAM. The other configurations have both SRAM and a 4-way set associative cache of various sizes. Mapped SRAM can be used for streaming video data and critical sections of code such as interrupt service routines. Cache is useful for most of the program and data structures.



**Figure 5. DM642 L1/L2 Cache**

## DM642 Enhanced DMA (EDMA) Controller

The DM642 EDMA can provide over 2Gbytes/sec of external bandwidth on initial implementations. The EDMA supports up to 64 channels triggered by independent events. A total of 85 parameter sets are available for linking or chaining. Linking allows a sequence of transfers to be issued when a single event occurs. Chaining allows one EDMA channel to trigger another channel upon data transfer completion. Linking and chaining allow continuous auto-initialization of DMA operation with only initial configuration by the CPU. These features also allow circular buffers, ping-pong buffers, and transfers of complex data structures. Transfers can be triggered on an element by element or frame by frame basis. Programmable triggering allows both sample by sample transfers and buffer by buffer transfers. Each channel supports both one and two-dimensional transfers. Strides are independently programmable for each dimension. Using 1-D and 2-D the user can transfer subframes of an image as well as automatically interleave or de-interleave time-division multiplexed (TDM) digital streams. Byte, half-word, word, and double-word data sizes are supported.

The EDMA supports unsurpassed concurrency. Four independent transfer queues allow highly efficient operation. Channels on different queues can interleave transfers on a cycle-by-cycle basis. For example, on cycle 1 queue 0 could service a DMA request triggered by one of the DM642 video ports to move data from the video port FIFO to external SDRAM through the EMIF. On cycle 2, the HPI could transfer data to mapped internal memory through queue 3. On cycle 3, audio data could be moved through queue 1 from internal memory to the McASP. On cycle 4, queue 2 could bring in data from the EMIF to internal memory requested by a user DMA. The key system benefit is that interactions between channels do not affect performance as much as much as in traditional DMA implementations.

## DM642 Video Port

The video port peripheral can operate as a video capture port, video display port, or TSI capture port. It provides the following functions:

- Video capture mode
  - Capture rate up to 80 MHz
  - Two channels of 8/10-bit digital video input from a digital camera or analog camera (using a video decoder). Digital video input is in YCbCr 4:2:2 format with 8 or 10 bit resolution multiplexed in ITU-R BT.656 format.
  - One channel of Y/C 16/20-bit digital video input in YCbCr 4:2:2 format on separate Y and Cb/Cr inputs. Supports SMPTE 260M, SMPTE 274M, SMPTE 296M, ITU-BT.1120, etc. as well as older CCIR601 interfaces.
  - YCbCr 4:2:2 to YCbCr 4:2:0 horizontal conversion and  $\frac{1}{2}$  scaling in 8-bit 4:2:2 modes
  - Direct interface for two channels of up to 10-bit or one channel of up to 20-bit raw video from A/D converters
- Video display mode
  - Display rate up to 110 MHz
  - One channel of continuous digital video output. Digital video output is YCbCr 4:2:2 co-sited pixel data with 8/10-bit resolution multiplexed in ITU-R BT.656 format.

- One channel of Y/C 16/20-bit digital video output in YCbCr 4:2:2 format on separate Y and Cb/Cr outputs (supports SMPTE 260M, SMPTE 274M, SMPTE 296M, ITU-BT.1120, etc.).
- YCbCr 4:2:0 to YCbCr 4:2:2 horizontal conversion and 2x scaling of output in 8-bit 4:2:2 modes
- Programmable clipping of BT.656 and Y/C mode output values
- One channel of raw data output up to 20-bits for interface to RAMDACs. Two channel synchronized raw data output.
- Sync to external video controller or another video display port
- Using the external clock, the frame timing generator provides programmable image timing including horizontal and vertical blanking, start of active video (SAV) and end of active video (EAV) code insertion and horizontal and frame timing pulses.
- Generates horizontal and vertical synchronization and blanking signals and a frame synchronization signal.
- TSI capture mode
  - Transport stream interface (TSI) from a front-end device such as demodulator or forward error correction device in 8-bit parallel format at up to 30 Mbytes/sec.
- The port generates up to three events per channel and one interrupt to the DSP.

## Video Port Architecture

A high-level block diagram of the video port is shown in Figure 6. As the diagram shows, the port consists of two channels – A and B. A 5120-byte capture/display buffer is splittable between the two channels. The entire port (both channels) is always configured for either capture or display only. Separate data pipelines control the parsing and formatting of capture or display data for each of the BT.656, Y/C, raw video, and TSI modes.

For capture operation, the port may operate as two 8/10-bit channels of BT.656 or raw video capture, or as a single channel of 8/10-bit BT.656, 8/10-bit raw video, 16/20-bit Y/C video, 16/20-bit raw video, or 8-bit TSI. For display operation, the port may operate as a single channel of 8/10-bit BT.656 display, or as a single channel of 8/10-bit BT.656, 8/10-bit raw video, 16/20 bit Y/C video, or 16/20-bit raw video display. It may also operate in a two channel 8/10-bit raw mode in which the two channels are locked to the same timing. Channel B is not used during single channel operation.

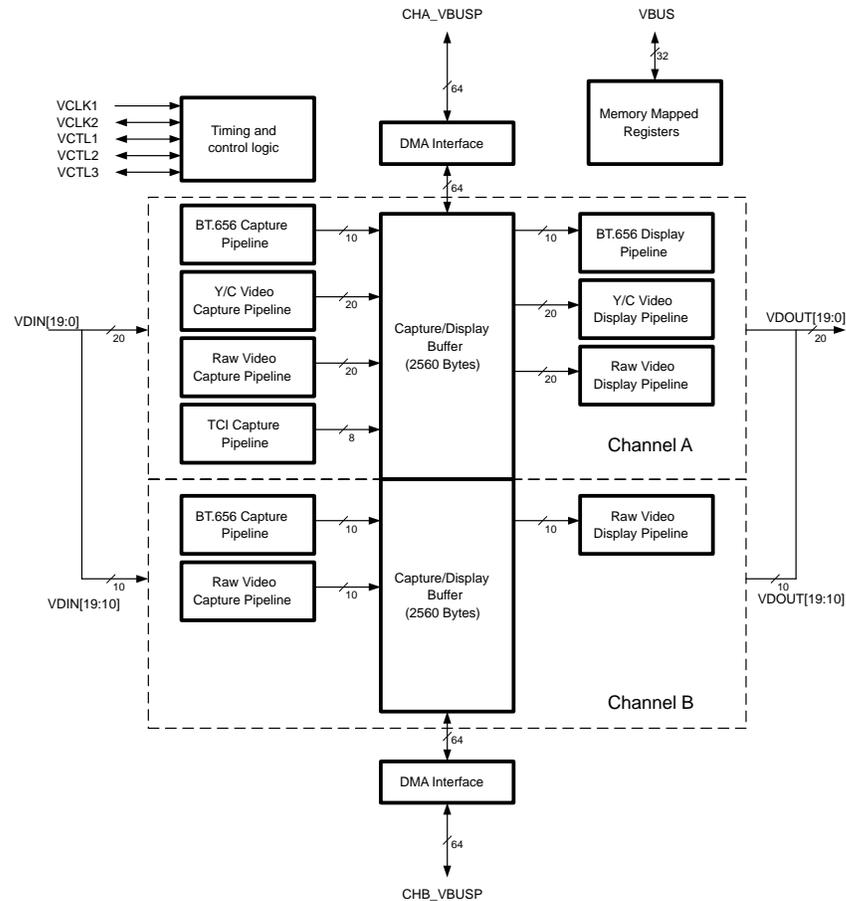


Figure 6. Video Port Block Diagram

## Video Port Signals

A video port uses 25 signals as listed below. These signals are referred to later on in this document in the video port explanation.

- One input clock and one input/output clock
  - VCLK1 – Channel A capture input clock/display input clock
  - VCLK2 – Channel B capture input clock/display output clock
- 20-bit data bus
  - VDATA[19:0] – Capture data input/display data output
- 3 Control Inputs
  - VCTL1 – Capture control input/display control output or input
  - VCTL2 – Capture control input/display control output or input
  - VCTL3 – Capture control input/display control output or input

## Video Capture Operation

Video capture works by sampling video data on the input pins and saving it to the video port FIFO. When the amount of captured data reaches a programmed threshold level, a DMA is performed to move data from the FIFO into DSP memory. In some cases, color separation (YC demultiplexing) is performed on the incoming video data requiring multiple FIFOs and DMAs to be used.

The port enables capture of both interlaced and progressive scan data. Interlaced capture can be performed on either a field-by-field or a frame-by-frame basis. A capture window specifies the data to be captured within each field. Frame and field synchronization can be performed using embedded sync codes or configurable control inputs allowing glueless interface to various encoders and ADCs.

The video capture module operates in one of nine modes as per Table 3.

**Table 3. Video Capture Modes**

Function	Description
8-bit ITU-R BT.656 capture	Digital video input is in <i>YCbCr</i> 4:2:2 with 8-bit resolution multiplexed in ITU-R BT.656 format.
10-bit ITU-R BT.656 capture	Digital video input is in <i>YCbCr</i> 4:2:2 with 10-bit resolution multiplexed in ITU-R BT.656 format.
8-bit raw capture	Raw 8-bit data capture at sampling rates up to 80 MHz.
10-bit raw capture	Raw 8-bit or 10-bit data capture at sampling rates up to 80 MHz.
8-bit Y/C capture	Digital video input is in <i>YCbCr</i> 4:2:2 with 8-bit resolution on parallel Y and Cb/Cr multiplexed channels.
10-bit Y/C capture	Digital video input is in <i>YCbCr</i> 4:2:2 with 10-bit resolution on parallel Y and Cb/Cr multiplexed channels.
16-bit raw capture	Raw 16-bit data capture at sampling rates up to 80 MHz.
20-bit raw capture	Raw 20-bit data capture at sampling rates up to 80 MHz.
TSI capture	8-bit parallel TSI capture at rates up to 30 MHz.

When operating as a raw video capture channel, no data selection or data interpretation is performed. The 16/20-bit raw capture mode is designed to accept data from A/D converters with resolution higher than eight bits (used, for example, in medical imaging).

### BT.656 Video Capture Mode

The BT.656 capture mode captures 8 or 10-bit 4:2:2 luma and chroma data multiplexed into a single data stream. Video data is conveyed in the order Cb,Y,Cr,Y,Cb,Y,Cr, etc. where the sequence Cb,Y,Cr refers to co-sited luma and chroma samples and the following Y value corresponds to the next luminance sample. The data stream is demultiplexed and each component is written in packed form into separate FIFOs for transfer into Y, Cb, and Cr buffers in DSP memory. (This is commonly called *planar* format.) The packing and order of the samples is determined by the sample size (8-bit or 10-bit) and the selected endianness of the DSP.

The ITU-BT.656 standard provides for either 8-bit or 10-bit component samples. When 10-bit samples are used, the 2 least significant bits are considered fractional values. Thus for 8-bit operation, input data is aligned to the most significant bits (9:2) of the input and the 2 LS bits are ignored.

In dual channel operation, the video port can capture two BT.656 data streams or one BT.656 data stream and one raw data stream. In the latter case, the BT.656 stream may occur on either Channel A or Channel B. If the port is configured for single channel operation, capture will take place on Channel A only.

The following operations may be optionally enabled with BT.656 capture:

- Chrominance Resampling: This is the horizontal component of Y/C 4:2:2 to Y/C 4:2:0 conversion. This operation is available only for 8-bit data.
- Horizontal  $\frac{1}{2}$  Scaling: This is useful for converting captured image data to lower resolutions (e.g CIF) for further processing. This operation is available only for 8-bit data.
- The video port allows capture of a user-defined window of BT.656 data per field. The user can specify start and stop pixel locations of the capture window. The captured window can be larger or smaller than the active video region.
- Programmable field or frame capture: This is useful for instance for capturing only even fields, only odd fields, single field/frame captures.

An example of BT.656 capture is shown in Figure 7. Analog video inputs from two cameras are fed into two TVP5145 NTSC/PAL decoders (for further information on TVP5145 please see <http://www.ti.com>). In this example, 10-bit BT.656 data from the TVP5145 devices is gluelessly input to the DM642.

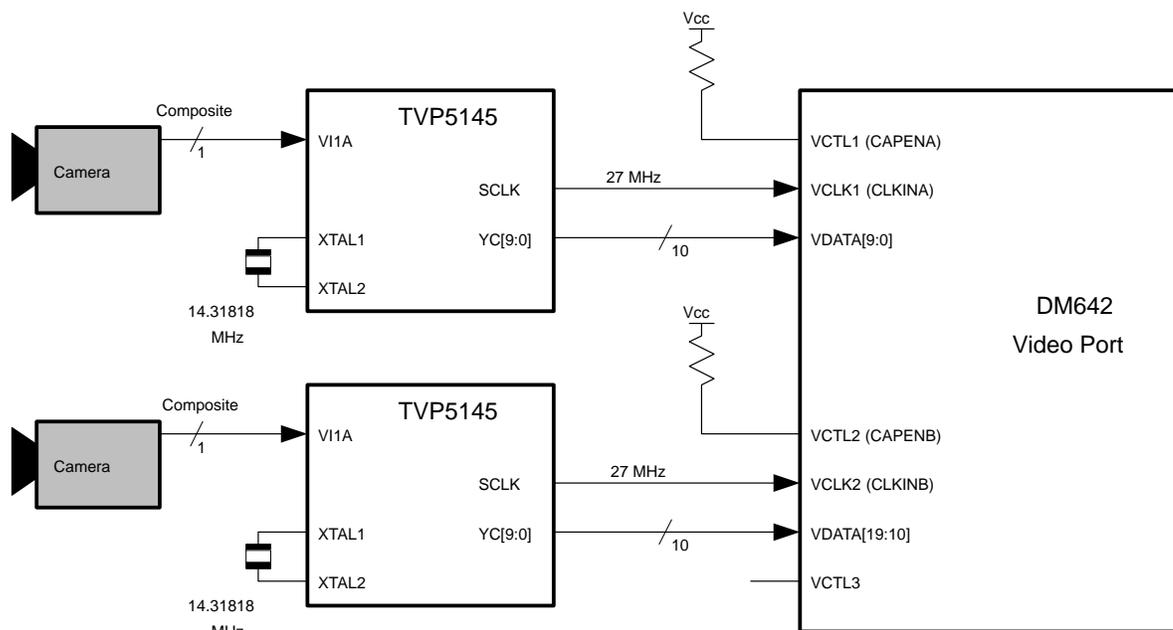


Figure 7. BT.656 Video Capture

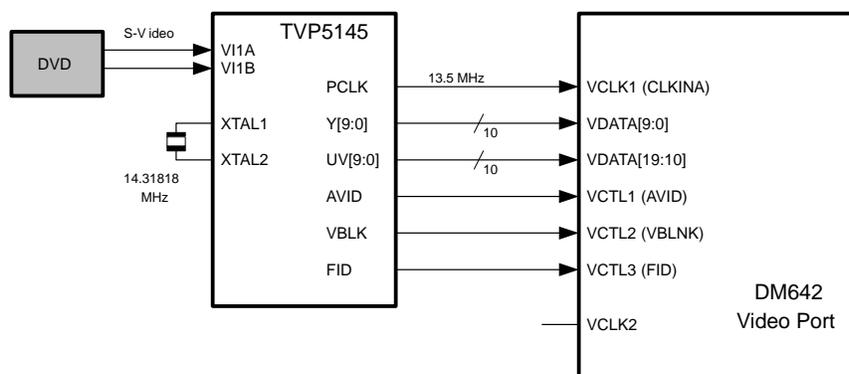
### Y/C Video Capture Mode

The Y/C capture mode supports HDTV standards such as SMPTE260, SMPTE296, and BT.1120 with embedded EAV/SAV codes. It also supports SDTV YCbCr modes that use separate control signals (sometimes called CCIR601 mode). Because 16 or 20 bits are used for data capture, the Y/C capture mode requires both halves of the video port data bus.

The following operations may be optionally enabled with Y/C capture:

- Chrominance Resampling: This is the horizontal component of Y/C 4:2:2 to Y/C 4:2:0 conversion. This operation is available only for 8-bit data.
- Horizontal  $\frac{1}{2}$  Scaling: This is useful for converting captured image data to lower resolutions (e.g. CIF) for further processing. This operation is available only for 8-bit data.
- The video port allows capture of a user-defined window of Y/C data per field (or per frame for progressive Y/C formats.) The user can specify start and stop pixel locations of the capture window. The captured window can be larger or smaller than the active video region.
- Programmable field or frame capture: This is useful for instance for capturing only even fields, only odd fields, single field/frame captures.

An example of Y/C capture is shown in Figure 8. Analog video from a DVD player is fed to the TVP5145. NTSC/PAL decoder (for further information on TVP5145 please see <http://www.ti.com>). In this example, digital YUV data at 10-bits per components is gluelessly interfaced from the TVP5145 to the DM642.



**Figure 8. Y/C Video Capture**

## Ancillary Data Capture

The BT.656 and some Y/C specifications includes provision for carrying ancillary (non-video) data within the horizontal and vertical blanking regions. Horizontal ancillary (HANC) data appears between the EAV code and SAV codes. Vertical ancillary data (VANC), also called vertical blanking interval (VBI) data appears during the active horizontal line portion of vertically blanking (e.g. after an SAV with V=1). Ancillary data blocks are always preceded by an ancillary data header of 0x00, 0xFF, 0xFF.

No special provisions are made for the capture of HANC data. HANC data can be captured using the normal video capture mechanism by programming the start of the capture window to occur before the SAV or by programming the stop of the capture window to occur past the EAV code. (The user must disable scaling and chroma resampling when including the capture of HANC data to prevent data corruption).

VANC or VBI data is commonly used for such features as teletext and closed-captioning. No special provisions are made for the capture of VBI data. VBI data may be captured using the normal capture mechanism by programming the start of the capture window to occur before the first line of active video on the first line of desired VBI data. The user must not enable scaling or chroma re-sampling when the capture of VBI data is desired or the data will be corrupted by the filters.

## Raw Video Capture Modes

This mode is useful for video data capture from A/D converters. Data is captured at the rate of the sender's clock, without any interpretation or start/stop of capture based on the data values.

8-bit, 10-bit, 16-bit, or 20-bit words are conveyed as a single parallel data stream at up to 80 Mwords/sec. In case of 8-bit or 10-bit capture, two streams may be simultaneously captured using the two channels of a video port. In case of 16-bit or 20-bit data, one stream may be captured per video port. Video filtering and demultiplexing operations are disabled.

## TSI Capture Mode

The TSI capture mode captures MPEG-2 transport data. The transport stream can carry multiple audio, video, data program streams, all multiplexed into one transport stream. The MPEG-2 transport stream uses fixed length packets which offers a high level of flexibility in allocating channel capacity among video, audio, and data services. The MPEG-2 system makes use of a packet identification (PID) field in the packet header as a means of data identification, and this allows a mix of video, audio, and data, which need not be specified in advance.

The data packet is divided into header, adaptation field, and data payload regions. The header is always 4 bytes long, and consists of a SYNC byte, PID, error indicator, priority level, scrambling control, payload start indicator, and adaptation field control. The adaptation field is an optional field, and carries the program clock reference (PCR). The data payload can be sent scrambled, or unscrambled.

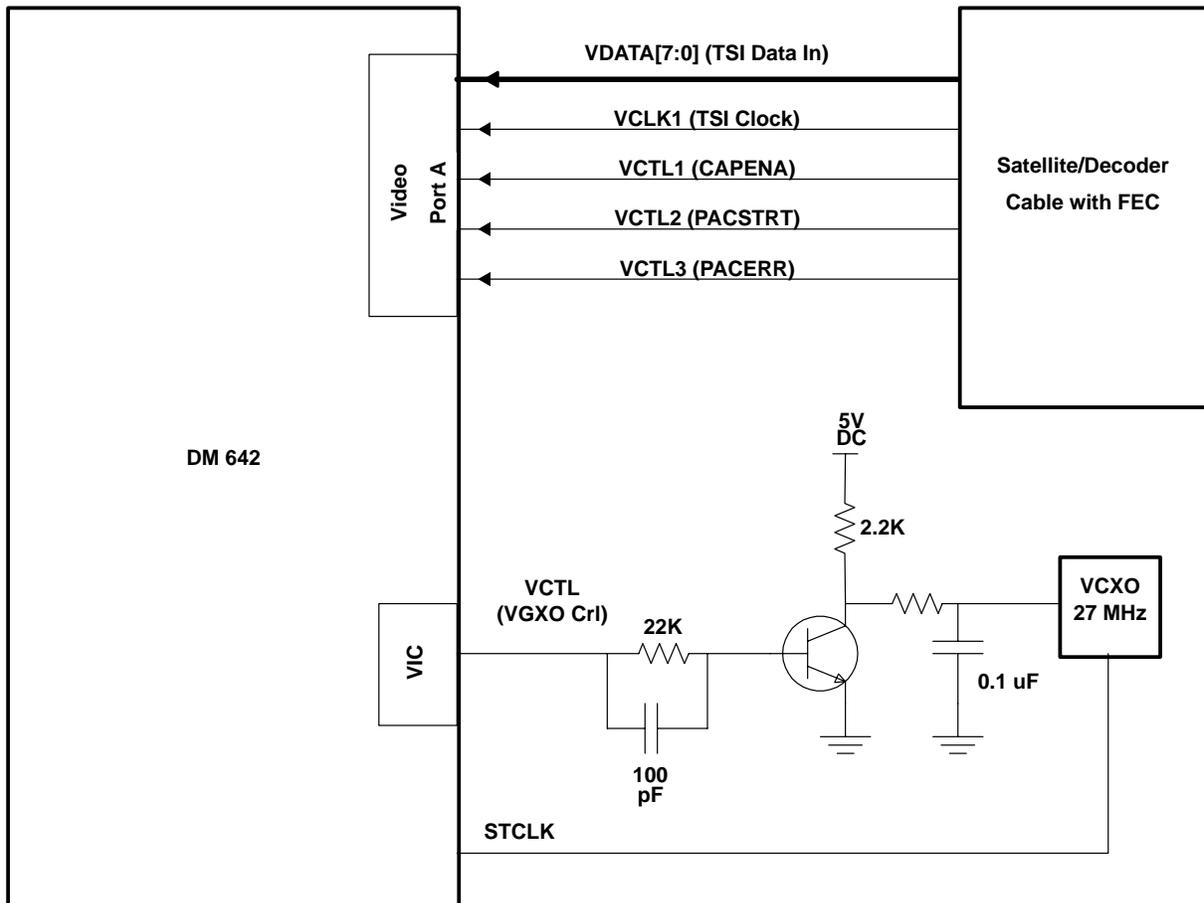
A typical decoder system performs the functions of capturing the data, detecting the SYNC byte, filtering the data based on the PID field, parsing the data for control information such as adaptation fields, de-scrambling the data, etc. The video port TSI capture mode supports the following features:

- Supports SYNC detect
- Data capture at the rising edge of incoming clock
- Parallel data reception
- Maximum data rate of 30 Mbytes/sec
- Programmable packet size
- Hardware counter mechanism to timestamp incoming packet data
- Programmable filtering of packets with errors
- Interrupt to the DSP based on absolute system time, or system time clock cycles

The video port does not perform the following functions. These functions should be performed in software:

- PID filtering; Data parsing; De-scrambling of data

An example of MPEG-2 transport stream capture is shown in Figure 9.



**Figure 9. MPEG-2 Transport Stream Capture**

## Video Input Filtering

The video input filter performs hardware scaling and resampling on incoming 8-bit BT.656 or 8-bit Y/C data. Filtering hardware is always disabled during 10-bit or raw data capture modes.

The input filter has four modes of operation:

- No filtering mode
- Chrominance resampling mode
- ½ Scaling mode
- ½ Scaling with chrominance resampling mode

## Chrominance Resampling

Chrominance resampling performs the horizontal portion of a conversion between YCbCr 4:2:2 format and YCbCr 4:2:0 format. This filter computes chrominance values at sample points midway between the input luminance samples based on the input co-sited chrominance samples. The vertical portion of the conversion must be performed in software.

The chrominance resampling filters calculate the implied value of Cb and Cr in between luminance sample points based upon nearby co-sited Cb and Cr samples. The resulting values are clamped to between 0x01 and 0xFE and sent to the Cb and Cr capture buffers. Chrominance re-sampling is shown in Figure 10.

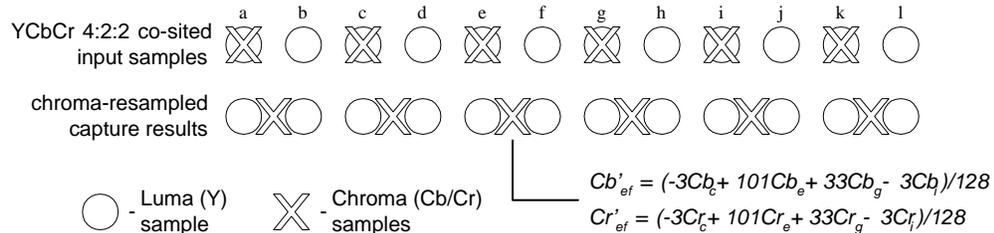


Figure 10. Video Input Chrominance Resampling

### 1/2 Scaling

The 1/2-scaling mode is used to reduce the horizontal resolution of captured luminance and chrominance data by a factor of two. Vertical scaling must be performed in software. For applications that require only CIF or lower resolutions, this reduces the video capture buffer memory requirements (and the bandwidth needed to write the buffer) by a factor of two. It also helps to save CPU cycles and reduces software development.

The filtering for the luminance portion of the scaling filter changes depending on whether chrominance re-sampling is also enabled. (By changing the luminance filter, the chrominance filters can remain the same.) The resulting values are clamped to between 0x01 and 0xFE and sent to the Y, Cb, and Cr capture buffers. Scaling for co-sited capture is shown in Figure 11 and scaling for chrominance re-sampling is shown in Figure 12.

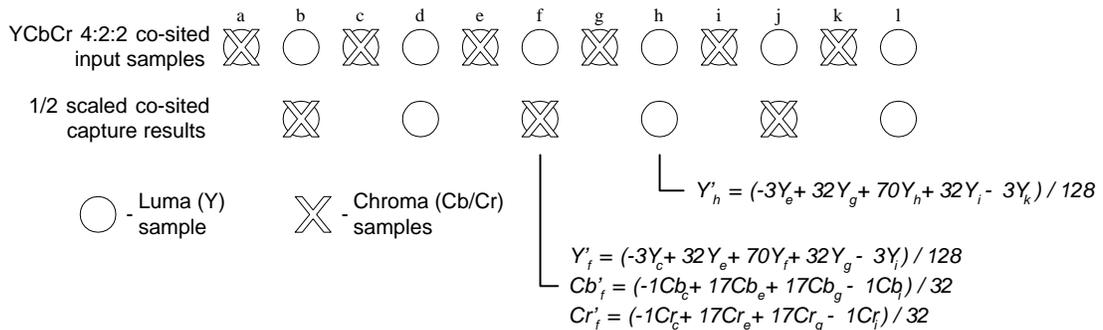
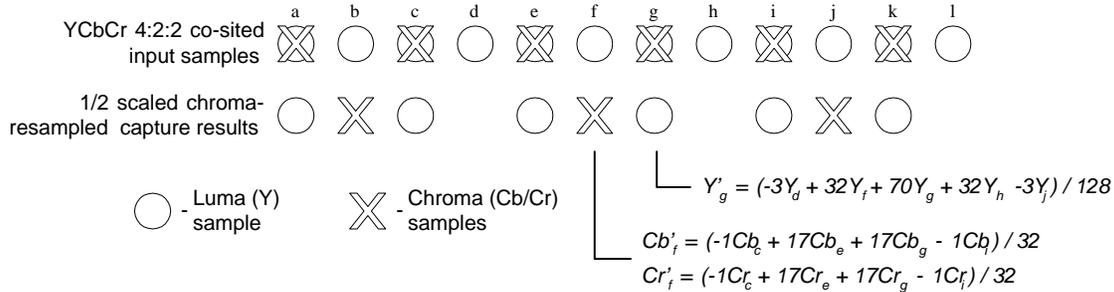


Figure 11. Video Input 1/2 Scaling

Note that because input scaling is limited to 1/2, true CIF horizontal resolution is not achieved if the full BT.656 horizontal line (720 pixels) is captured. A CIF size line can be captured by selecting a 704 pixel-sized window within the BT.656 line and subjecting it to 1/2 scaling.

Because the filters make use of preceding and trailing samples, filtering artifacts can occur at the beginning of the BT.656 or Y/C active line because no samples exist before the SAV code, and at the end of the BT.656 active line because no samples exist after the EAV code. In order to minimize artifacts, the first  $m$  samples after sample 0 (where  $m$  is the maximum number of preceding samples used by any of the filters) are mirrored to the left of sample 0 and the last  $m$  samples before the last sample are mirrored to the right of the last sample.



**Figure 12. Video Input 1/2 Scaling With Chrominance Resampling**

### Video Display Operation

Video Display can operate in one of eight modes listed in Table 4.

**Table 4. Video Display Modes**

Function	Description
8-Bit ITU-R BT.656 display	Digital video output is in YCbCr 4:2:2 with 8-bit resolution multiplexed in ITU-R BT.656 format
10-Bit ITU-R BT.656 display	Digital video output is in YCbCr 4:2:2 with 10-bit resolution multiplexed in ITU-R BT.656 format
8-Bit Raw display	8-bit data output
10-Bit Raw display	10-bit data output
8-Bit Y/C display	Digital video is output in YCbCr 4:2:2 with 8-bit resolution on parallel Y and Cb/Cr multiplexed channels
10-Bit Y/C display	Digital video is output in YCbCr 4:2:2 with 10-bit resolution on parallel Y and Cb/Cr multiplexed channels
16-Bit Raw display	16-bit data output
20-Bit Raw display	20-bit data output

### BT.656 Video Display Mode

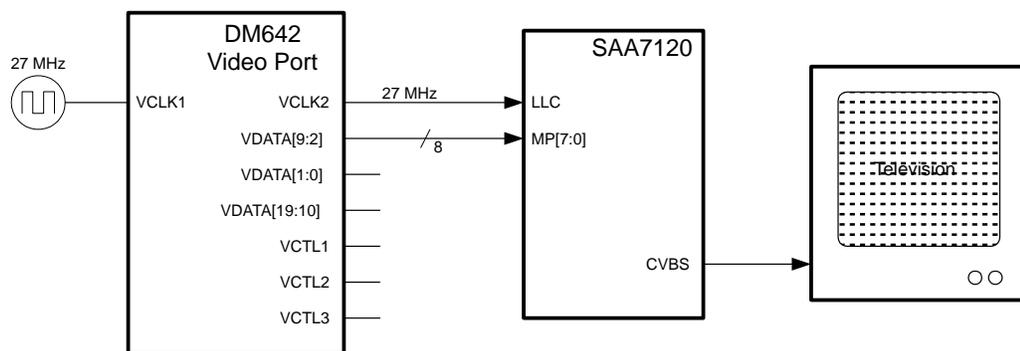
The BT.656 display mode outputs one channel of Y/C 8-bit or 10-bit 4:2:2 co-sited luma and chroma data multiplexed into a single data stream. Pixels are output in pairs, with each pair consisting of two luma samples and two chroma samples. The chroma samples are associated with the first luma pixel of the pair.

The following operations may be optionally enabled with BT.656 display:

- Chrominance Resampling: This is the horizontal component of Y/C 4:2:0 to Y/C 4:2:2 conversion. This operation is available only for 8-bit data.

- Horizontal 2x Scaling: This is useful for converting image data from lower resolutions (e.g CIF) to higher resolutions for display. This operation is available only for 8-bit data.
- The video port allows display of a user-defined window of BT.656 data per field. The user can specify start and stop pixel locations of the capture window.
- Programmable clipping of BT.656 output values.
- Using the external clock, the frame timing generator provides programmable image timing including horizontal and vertical blanking, SAV and EAV code insertion, and horizontal and frame timing pulses.

Figure 13 shows an example of BT.656 display. In this example, 8-bit BT.656 data is gluelessly interfaced from the DM642 to the SAA7120, a video encoder device.



**Figure 13. BT.656 Display**

## Y/C Video Display Mode

The Y/C display mode is similar to the BT.656 display mode but outputs 8 or 10-bit data on separate luma and chroma data streams. One data stream contains Y samples and the other stream contains multiplexed Cb and Cr samples co-sited with every other luminance sample. The Y samples are read from the Y FIFO and the Cb and Cr samples are read from the Cb and Cr FIFOs and combined on the chroma output. The unpacking and order of the samples is determined by the sample size (8 or 10-bit), and the selected endianness of the device.

The Y/C display mode can generate HDTV standard output such as BT.1120, SMPTE260, or SMPTE296 with embedded EAV and SAV codes. It can also output separate control signals.

Because 16 or 20 bits are used for data output, the Y/C output mode requires both halves of the video port data bus.

The following operations may be optionally enabled with Y/C display:

- Chrominance Resampling: This is the horizontal component of Y/C 4:2:0 to Y/C 4:2:2 conversion. This operation is available only for 8-bit data.
- Horizontal 2x Scaling: This is useful for converting image data from lower resolutions (e.g CIF) to higher resolutions for display. This operation is available only for 8-bit data.
- The video port allows display of a user-defined window of Y/C data per field (or per frame for progressive Y/C formats.) The user can specify start and stop pixel locations of the display window.

- Programmable clipping of Y/C output values.
- Using the external clock, the frame timing generator provides programmable image timing including horizontal and vertical blanking, SAV and EAV code insertion, and horizontal and frame timing pulses.

## Ancillary Data Display

The BT.656 and some Y/C specifications includes provision for carrying ancillary (non-video) data within the horizontal and vertical blanking regions. Horizontal ancillary (HANC) data appears between the EAV code and SAV codes. Vertical ancillary data (VANC), also called vertical blanking interval (VBI) data appears during the active horizontal line portion of vertically blanking (e.g. after an SAV with V=1). Ancillary data blocks are always preceded by an ancillary data header of 0x00, 0xFF, 0xFF.

No special provisions are made for the display of HANC or VANC data. HANC data can be displayed using the normal video capture mechanism by programming the video port parameters to output data in the horizontal and vertical blanking regions. The HANC data including the ancillary data header must be part of the YCbCr separated data in the FIFOs. VBI data must also be YCbCr separated. The user must disable scaling and chroma re-sampling when including the display of HANC data or VBI data to prevent data corruption.

## Raw Video Display Mode

The raw data display modes are intended to output data to a RAMDAC or other D to A type device. This is typically RGB formatted data. No timing information is inserted into the output data stream; instead, selectable control signals are output to indicate timing. Raw data display includes a synchronized dual channel option. This allows Channel B to output a separate data stream using the same clock and control as Channel A. This mode is useful when used with a second video port in systems that require 24 – 30 bit RGB output.

Figure 14 shows an example of 30-bit RGB display using 2 video ports. In this example, RGB data at 10-bits per component is gluelessly interfaced from the DM642 to the THS8200, a triple 10-bit all format video DAC (for further information on THS8200 please see <http://www.ti.com>).

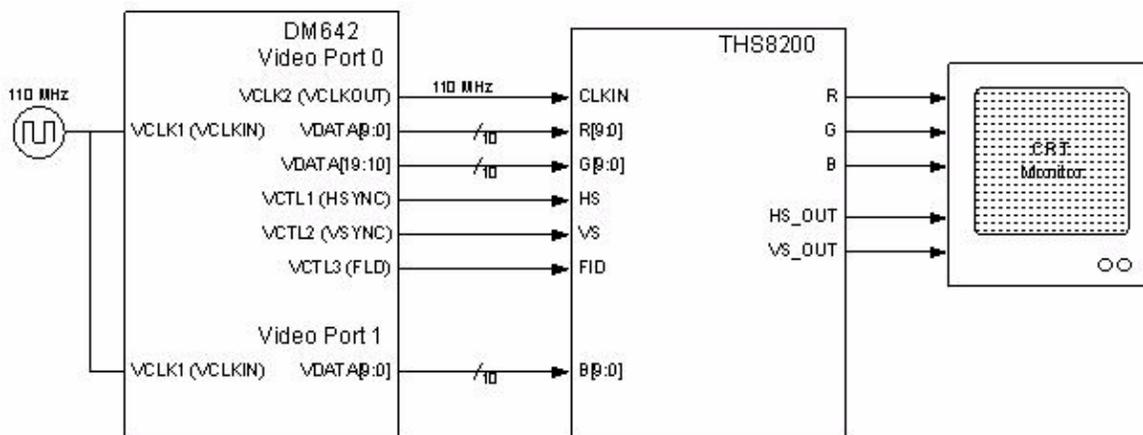


Figure 14. RGB Display

## Video Output Filtering

The video output filter performs simple hardware scaling and re-sampling on outgoing 8-bit BT.656 or Y/C data. Filtering hardware is disabled during 10-bit or raw data display modes.

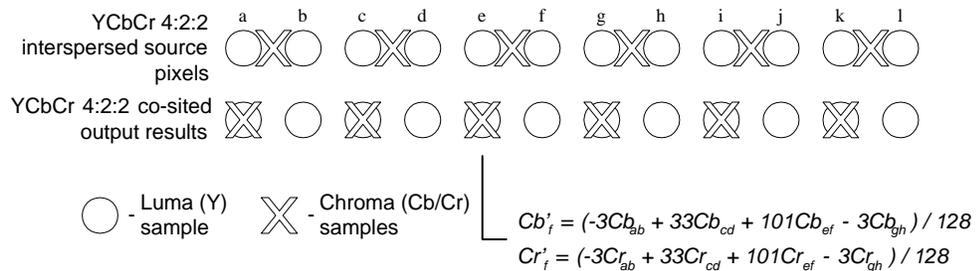
The output filter has four modes of operation:

- No filtering mode
- 2x scaling mode
- Chrominance resampling mode
- 2x scaling with chrominance resampling mode

## Chrominance Resampling Operation

Chrominance resampling performs the conversion between interspersed YCbCr 4:2:2 format and co-sited YCbCr 4:2:2 format. This filter computes chrominance values at sample points corresponding to output luminance samples based on the input interspersed chrominance samples. The vertical portion of the conversion from YCbCr 4:2:0 to interspersed YCbCr 4:2:2 must be performed in software.

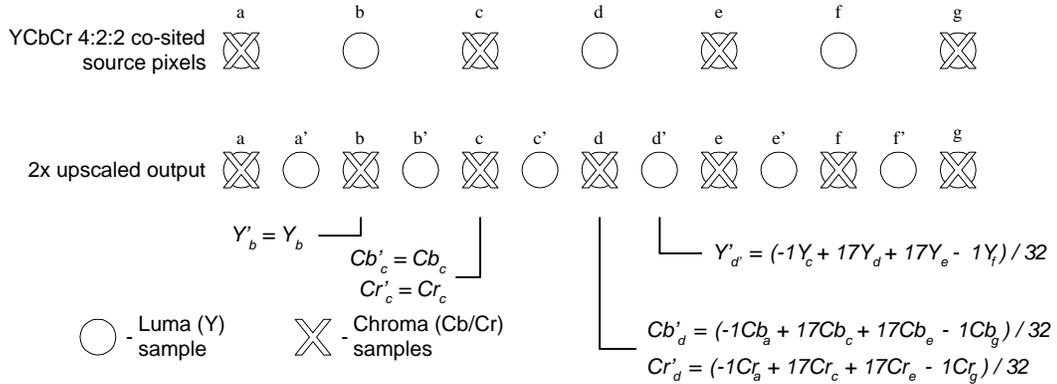
The chrominance resampling filters calculate the implied value of Cb and Cr co-sited with luminance sample points based upon nearby interspersed Cb and Cr samples. The resulting values are clamped to between 0x01 and 0xFE before being output. Chrominance resampling is shown in Figure 15.



**Figure 15. Video Output Chrominance Resampling**

## 2x Scaling Operation

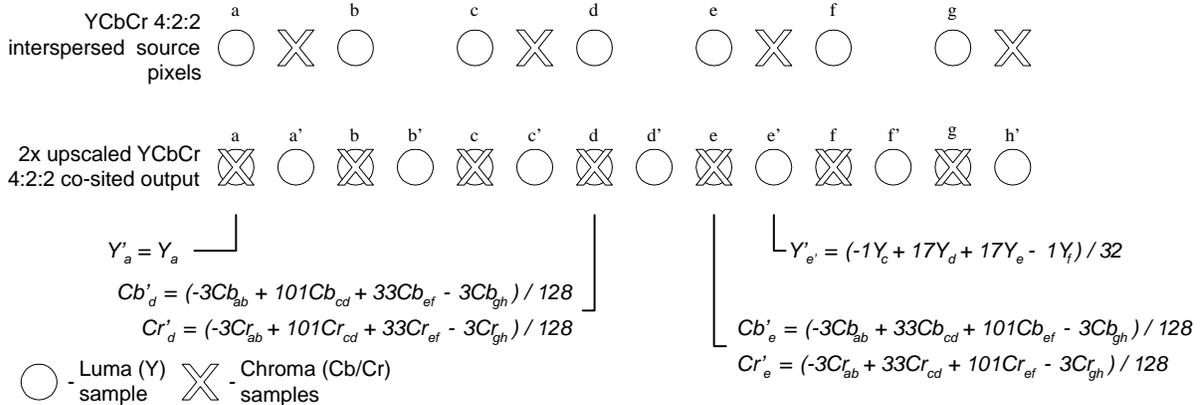
The 2x scaling mode is used to double the horizontal resolution of output luminance and chrominance data. This allows processed CIF resolution images to be output at full size. Vertical scaling must be performed in software. Scaling for co-sited source is shown in Figure 16 and scaling for interspersed source is shown in Figure 17.



**Figure 16. Video Output 2x Scaling**

For a co-sited source, the source luminance pixels are output unchanged for every even pixel (a, b, c, etc in Figure 16). Odd luminance pixels (a', b', c', etc) are generated from neighboring source (even) pixels using a four tap filter. The chrominance source pixels are output unchanged for every other even pixel (a, c, e, etc). Other even output pixel (b, d, f, etc) chrominance values are generated from neighboring source chrominance pixels using a four tap filter.

For an interspersed source, the luminance is output identically to the co-sited case. Chrominance output is generated using a 4-tap filter with one of two different coefficient sets.



**Figure 17. Video Output 2x Scaling With Chrominance Resampling**

Note that because input scaling is limited to 2x, full BT.656 width output is not achieved from CIF source images. As four tap filters are used on the output, the first and last two pixels on each line are mirrored in the video port to avoid filtering edge effects.

## VCXO Interpolated Control Port (VIC)

The VCXO interpolated control port (VIC) provides digital to analog conversion with resolution from 9 bits to up to 16 bits. The output of the VIC is a single bit interpolated D/A output.

Typical D/A converters provide a discrete output level for every value of the digital word that is being converted. This is a problem for digital words that are long. This is avoided in a Sigma Delta type D/A converter by choosing a few widely spaced output levels and interpolating values between them. The interpolating mechanism causes the output to oscillate rapidly between the levels in such a manner that the average output represents the value of input code.

In the VIC, two output levels are chosen (0 and 1), and Sigma Delta interpolation scheme is implemented to interpolate between these levels with a rapidly changing signal. The frequency of interpolation is dependent on the resolution needed.

The VIC supports the following features:

- Single interpolation for D/A conversion
- Programmable precision from 9 to 16 bits
- Interface for register accesses

The VIC port can be used to control system clock VCXO for MPEG transport stream interface.

Synchronization is an important aspect of decoding and presenting data in real-time digital data delivery systems. This is addressed in the MPEG transport packets by transmitting timing information in the adaptation fields of selected data packets. This serves as a reference for timing comparison in the receiving system. A sample of the 27 MHz clock, the program clock reference (PCR) shown in Figure 18, is transmitted within the bit stream, which indicates the expected time at the completion of reading the field from the bit stream at the transport decoder. The sample is a 42-bit field 9 bits of which cycle from 0 to 299 at 27 MHz, while the other 33-bit field is incremented by 1 each time the 9-bit field reaches a value of 299. The transport data packets are in sync with the server system clock



**Figure 18. PCR Header Format**

The video port can, in conjunction with the VIC Port, use a combined hardware and software solution to synchronize the transport system time clock (STC) with the clock reference transmitted in the bit-stream.

The video port maintains a hardware counter that counts the system time. The counter is driven by a system time clock (STCLK) input driven by an external VCXO controlled by VIC Port.

On reception of a packet, video port captures a snapshot of the counter. The software uses this timestamp to determine the deviation of the system time clock from the server clock, and drives VDAC output of the VIC Port to keep it synchronized.

Any time a packet with a PCR is received, the timestamp for that packet is compared with the PCR value in software. A PLL is implemented in software to sync up the local STCLK with the source system time clock. The DSP updates the VIC input register using the output from this algorithm, which in turn drives the VDAC output that controls the system time clock VCXO.

The user sets the precision for the VIC port. The frequency of PCR and precision desired determines the interpolation rate, which in turn decides the VIC clock divider.

## Ethernet Media Access Controller (EMAC)

The ethernet media access controller (EMAC) provides an efficient interface between the DSP core processor and the network. The DM642 EMAC supports both 10Base-T and 100Base-TX, or 10Mbits/second and 100Mbits/second in either half or full duplex, with hardware flow control and quality of service (QOS) support. The DM642 EMAC makes use of a custom interface to the DSP core that allows efficient data transmission and reception.

The following is the basic feature set of the DM642 EMAC:

- Standard media independent interface (MII) to physical layer device (PHY)
- Eight receive channels with VLAN tag discrimination for receive QOS support
- Eight transmit channels with round-robin or fixed priority for hardware QOS support
- Synchronous 10/100Mbit operation
- Ether-stats and 802.3-stats statistics gathering
- Transmit CRC generation selectable on a per channel basis
- Broadcast frames selection for reception on a single channel
- Multicast frames selection for reception on a single channel
- Promiscuous receive mode frames selection for reception on a single channel (all frames, all good frames, short frames, error frames)
- Hardware flow control

### Interface Feature Set

The EMAC interface includes the following features:

- EMAC acts as “DMA master” to either internal or external DSP memory space
- EMAC registers and descriptor memory mapped to configuration space on the DSP
- Local EMAC descriptor memory allows the EMAC to operate on descriptors without affecting the DSP. The 4K byte descriptor memory holds enough information to transfer up to 256 ethernet packets without DSP intervention.
- Programmable interrupt logic allows the ethernet driver to restrict the generation of back to back interrupts, allowing more work to be performed in a single call to the interrupt service routine.

### EMAC Operation

The EMAC operates somewhat independently of the DSP. It is configured and controlled by a register set that is mapped into DSP memory. Information about data packets is communicated by a 4K byte descriptor RAM that is local to the EMAC and mapped into DSP memory space. For transmit operations, each 16 byte descriptor can describe a packet or packet fragment in DSP internal or external memory. For receive operations, each descriptor represents a free packet buffer or buffer fragment. On both transmit and receive, an ethernet packet is allowed to span one or more memory fragments, represented by one 16 byte descriptor per fragment. In typical operation, there will only be one descriptor per receive buffer, while packets for transmit may still be fragmented, depending heavily on the system software architecture.

An interrupt is issued to the DSP whenever a transmit or receive operation is completed. However, it is not necessary for the DSP to service the interrupt while there are additional resources available. In other words, the EMAC will continue to receive ethernet packets until its receive descriptor list has been exhausted. On transmit, transmit descriptors need only be serviced to recover their associated memory buffer. In order to take advantage of this delayed servicing option, and to reduce interrupt generation to the DSP, the hardware is programmable as to the minimum time allowed for the generation of a back to back interrupt. This delays the subsequent interrupts in a traffic burst, allowing the DSP to process multiple packets per interrupt.

Eight channels are supplied for both transmit and receive operations. On transmit, the eight channels represent eight independent transmit queues. The EMAC can be configured to treat these channels as an equal priority round-robin queue, or as a set of eight fixed priority queues. On receive, the eight channels represent eight independent receive queues with packet classification. Packets are classified based on the destination MAC address. This can be unicast (one per channel), or by multicast, broadcast, or other (promiscuous, error, etc.).

The EMAC keeps track of 36 different statistics, and it tracks the status of each individual packet in its corresponding packet descriptor.

## Management Data Input/Output (MDIO)

The MDIO module continuously polls all 32 MDIO addresses in order to enumerate all PHY devices in the system. Once a PHY candidate has been selected by the DSP, the MDIO module transparently monitors its link state by reading the PHY status register. Link change events are stored on the MDIO device, which can optionally interrupt the DSP. This allows the DSP to poll the link status of the device without continuously performing costly MDIO accesses.

However, when the DSP must access MDIO for configuration and negotiation, the module performs the actual MDIO read or write operation independent of the DSP, allowing the DSP to poll for completion or interrupting the DSP once the operation has been performed.

Figure 19 shows DM642 networking using the EMAC and MDIO.

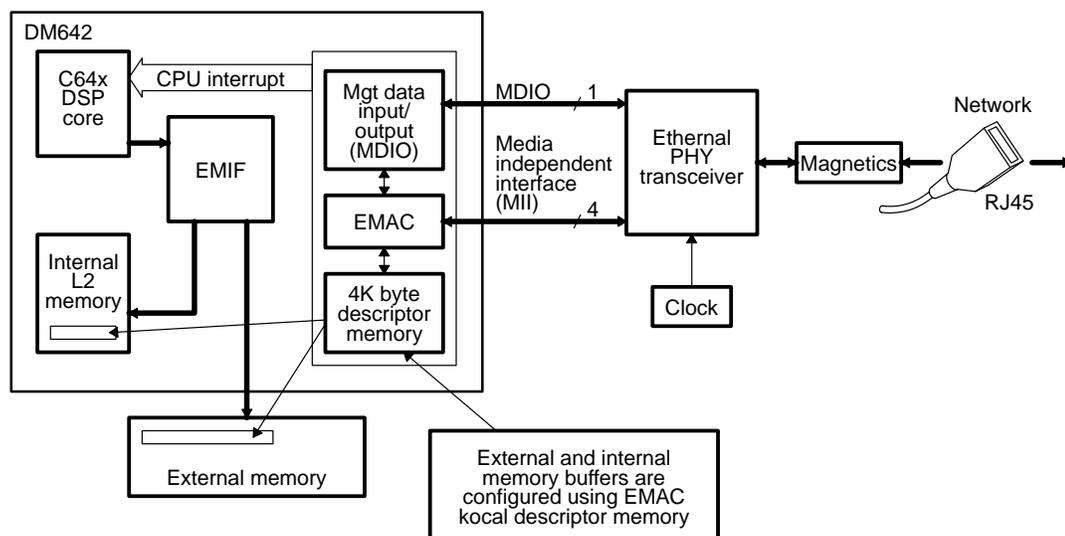


Figure 19. DM642 Networking Using EMAC and MDIO

## Multichannel Audio Serial Port (McASP)

The DM642 multichannel audio serial port (McASP) functions as a general-purpose audio serial port optimized for the needs of multichannel audio applications. The McASP is useful for both inter-integrated sound (IIS) protocols and inter-component digital audio interface transmission (DIT). The McASP consists of transmit and receive sections that may operate completely independently with separate master clocks, bit clocks, and frame syncs, and using different transmit modes with different bitstream formats. Alternatively, the transmit and receive sections may operate synchronized. In addition, all of the McASP pins can be configured as general-purpose input/output (GPIO) pins.

The McASP is intended to be flexible so that it may connect gluelessly to audio analog-to-digital converters (ADC), digital-to-analog converters (DAC), codec, digital audio interface receiver (DIR), and SPDIF transmit physical layer components. For DIR reception, an external DIR receiver integrated circuit should be used with IIS output format and connected to the McASP receive section.

Features of the McASP include

- Two independent clocks (transmit and receive)
- 8 serial data pins, individually assignable

Each clock includes

- Programmable clock generator
- Programmable frame sync generator
- TDM streams from 2 to 32, and 384 time slots
- Support for slot sizes of 8, 12, 16, 20, 24, 28, and 32 bits
- Data formatter for bit manipulation
- Wide variety of IIS and similar bit stream format

Integrated digital audio interface transmitter (DIT) supports

- SPDIF, IEC60958-1, AES-3 formats
- Up to 8 transmit pins
- Enhanced channel status/user data RAM
- Extensive error checking and recovery

## Other DM642 Peripherals

### External Memory Interface (EMIF)

The DM642 EMIF is 64-bits wide and is intended for direct connection to high-speed synchronous memory. On initial implementations, the EMIF has a maximum bus rate of 133 MHz. The EMIF has four chip enable (CE) spaces. It can support read and operations to 64-, 32-, 16-, and 8-bit external devices. The EMIF has three memory controllers. The SDRAM

controller supports 16 Mbit – 256 Mbit SDRAM devices. A programmable synchronous controller with selectable read/write latency offers direct connection to flow-through synchronous burst SRAMs, standard-write synchronous burst SRAMs, ZBT (zero bus turnaround) synchronous burst SRAMs, synchronous FIFOs, and clocked FIFOs. Finally, a programmable asynchronous controller with independent setup, strobe, and hold control allows easy interface to many asynchronous SRAMs, FIFOs, and peripheral devices. The EMIF operates with dedicated external clock input that decouples CPU operating frequency from bus frequency. In addition, particular controllers can operate at 1x, 1/2x or 1/4x the bus input clock. All these features are independently configurable for each CE space.

## Host Port Interface (HPI)

A 32-bit wide HPI provides dedicated connection to a variety of industry standard host processors and PCI bridge chips. The HPI can operate in either a 32-bit (HPI32) or 16-bit (HPI16) wide mode. An additional use of the HPI is as a slave port through which a mastering peripheral can stream data into the DSP. Through the HPI, the host controller has access to the DSP's entire memory map. The host can also access memory-mapped peripherals through the host interface.

## Peripheral Component Interconnect (PCI)

The 32-bit wide HPI is multiplexed with a PCI port. The PCI interface supports connection of the DM642 to a PCI host via the integrated PCI master/slave bus interface and features a 32-bit address/data bus at 66 MHz. The DM642 PCI interface contains the logic required to implement a fully compliant PCI specification revision 2.2 bursting master/slave with access into the DSP's memory map (peripherals, on-chip RAM, and external memory through the EMIF). The PCI interfaces to the DM642 via the EDMA internal address generation hardware. This architecture allows for both PCI master and slave transactions, while keeping the EDMA channel resources available for other applications.

## Multichannel Buffered Serial Ports (McBSPs)

Two multichannel buffered serial ports (McBSPs) interface to a variety of standards. McBSPs are synchronous serial ports. Each McBSP supports independent channel selection at any given time for up to 128-channels. The 128 channels represent a full ST-bus span. ST-bus in combination with the flexible asynchronous interface provides a glueless connection to a variety of multichannel telecommunications interface products such as H.110/H.100 framers as well as T1/E1 framing chips and the IOM2 bus. Multiple audio standards such as IIS and AC97 are directly supported allowing interface to stereo multichannel audio devices. Finally, the SPI mode allows connection to serial control devices and ROMs.

## General Purpose I/O Support

The general-purpose input/output (GPIO) peripheral provides dedicated general-purpose I/O support. When configured as an output, the user can control the state driven on the output pin. When configured as an input, the user can detect the state of the input, which is reflected in an internal register. While there are a total of 16 GPIO pins, some are multiplexed with other device pins. In addition, the GPIO peripheral can produce CPU interrupts and EDMA events in different interrupt/event generation modes.

## Additional Peripheral Information

The DM642 peripherals play an integral role in sustaining the system I/O bandwidth. For more detailed information on these new device interfaces and for specific peripheral information, please refer to the device data sheets and to the DM642 Peripherals Guide. These documents will be available at the following URL: <http://www.ti.com>.

## DM642 Peripherals Multiplexing

The DM642 includes a number of peripherals to support video and imaging applications. The peripherals are listed again below:

- Three configurable video ports capable of glueless video input, video output, or transport stream input.
- VCXO interpolated control port (VIC)
- 10/100 Mbps ethernet MAC (EMAC)
- Management data input/output (MDIO) module
- Multichannel audio serial port (McASP)
- Inter-Integrated circuit (I2C) bus module
- Two multichannel buffered serial ports (McBSPs)
- Three 32-bit general-purpose timers
- User-configurable 16-bit or 32-bit host-port Interface (HPI16/HPI32)
- 66 MHz 32-bit peripheral component interconnect (PCI)
- A general-purpose input/output (GPIO) port
- 64-bit external memory interface (EMIF) that is capable of interfacing to synchronous and asynchronous memories and peripherals

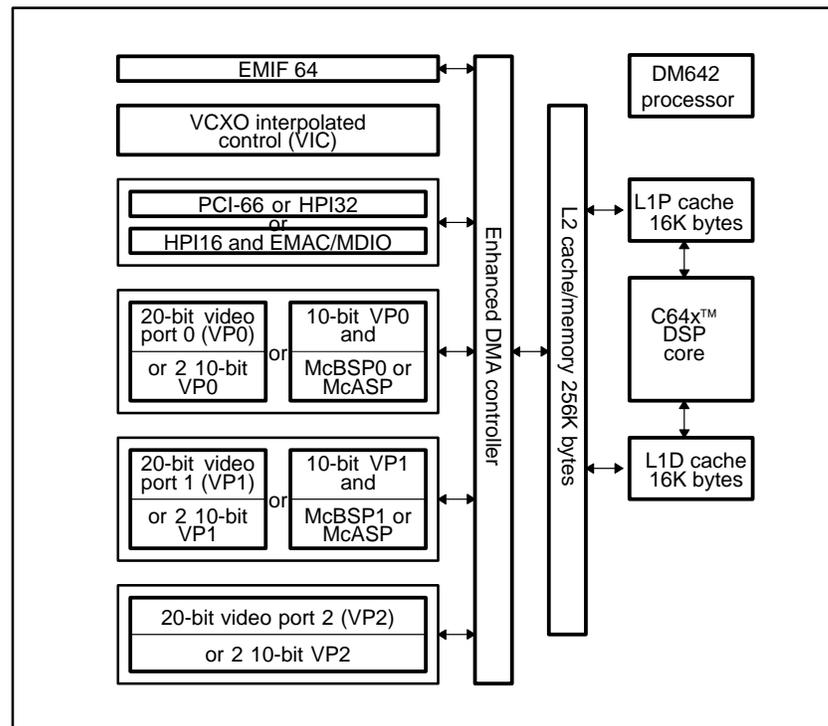
Several of these peripherals are multiplexed with one another and the appropriate set of peripherals must be selected by the user during processor configuration.

- The PCI interface is multiplexed with the 32-bit host port interface (HPI), and with a combination of 16-bit HPI and EMAC/MDIO. This provides the following options to the user:
  - 32-bit 66 MHz PCI bus
  - 32-bit HPI
  - Combination of 16-bit HPI and EMAC/MDIO.
- The first video port (VP0) provides the following options:
  - Configurable as a single video port upto 20-bits wide
  - Configurable as two video ports, each upto 10-bits wide
  - Configurable as a video port upto 10-bits wide and a McBSP
  - Configurable as a video port upto 10-bits wide and McASP (control signals)

- The second video port (VP1) provides similar options:
  - Configurable as a single video port upto 20-bits wide
  - Configurable as two video ports, each upto 10-bits wide
  - Configurable as a video port upto 10-bits wide and a McBSP
  - Configurable as a video port upto 10-bits wide and McASP (data signals)
- The third video port (VP2) provides the following options:
  - Configurable as a single video port upto 20-bits wide
  - Configurable as two video ports, each upto 10-bits wide

Note: If using McASP, both VP0 and VP1 must be configured in McASP mode.

Figure 20 illustrates the various peripheral multiplexing options available on DM642.



**Figure 20. DM642 Peripherals Multiplexing**

## DM642 Application Examples

The DM642 digital media processor can be used in a wide variety of video and imaging applications. Some example applications are described here.

## Video IP Phone

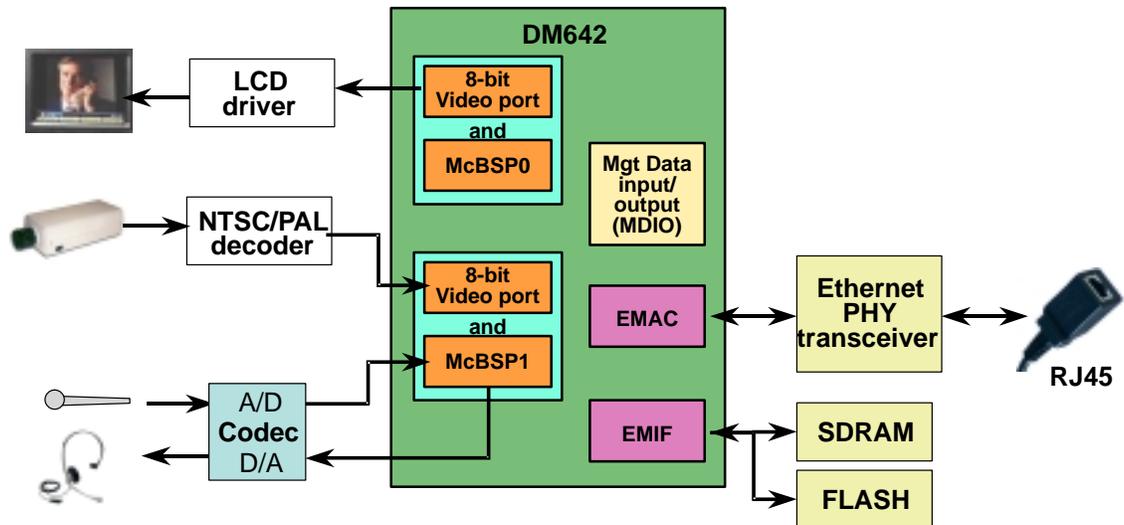
Traditional phone circuits are based on circuit switched networks. An IP phone replaces the circuit switched network backbone by utilizing data packet switching to transport voice. The IP phone is a desktop business telephone that delivers enhanced, low cost telephony services by leveraging the corporate data networking infrastructure (LAN). A video IP phone is an enhancement to IP phones wherein video and voice data are transmitted over the same network. A video IP phone design requires the following components:

- Voice interface (microphone for input/speaker for output)
- Video interface (camera interface for capture/LCD for display)
- Network interface
- User interface (phone keypad)
- Engine to compress voice and video data

On the transmit side, video data is captured from a camera interface, digitized and sent to a compression engine. The compressed video data is sent to the remote end over the network. The video data from remote users is received via the network interface, decompressed and displayed on the LCD panel. The display also shows user prompt information, caller ID (if supported), number dialed etc. If conferencing mode is supported, the display may optionally accommodate multiple images. Voice interface does the conversion of analog voice into digital signals. Voice from the microphone sampled at 8kHz is digitized to a 64kbps PCM (pulse code modulated) stream and sent to the processor block for compression. Voice data from the remote user is decompressed by the processor, converted to analog and sent to the speaker on the phone. The user interface in the IP Phone is designed to facilitate keypad dialing and selection of other value added features on the phone like speakerphone, conferencing, mute, directory service etc. The network interface enables the transmission of the packetized data (voice and/or video) to the remote side. Some IP phones provide for an optional RJ45 jack to enable simultaneous connection to the PC and to the LAN.

Traditionally, the IP phone processing unit is composed of a digital signal processor to handle the signal processing functions and a micro controller to handle the network interface, user interface etc. The TMS320DM642 is a digital media processor that at 600 MHz delivers processing power of 4800+ MIPS. DM642 has an advanced VelocityTI.2 architecture with special instructions that accelerate video data processing by further utilizing the parallelism inherent in the architecture. In addition to all the processing power, the DM642 also has new improved peripherals for better system integration in cost sensitive applications like video IP phones.

TMS320DM642 has video ports that can capture or display BT656 or raw video data at high speeds. This provides glue-less interface to NTSC/PAL encoder and decoder chips. The device supports the McASP or McBSP peripheral that enables hook-up to a variety of analog-to-digital or digital-to-analog converters (ADC/DAC) for microphone and speaker interface. The external memory interface (EMIF) on the DM642 offers a high-speed high bandwidth interface to different memory types both asynchronous and synchronous. The seamless interface to synchronous memories especially high density SDRAMs enable storage of large video data buffers at very little system cost. The EMIF also supports a glueless interface to non-volatile memories to initialize the system. This DSP also has a 10/100Mbps ethernet MAC that through an ethernet PHY device provides connectivity to the network for packet interface. Figure 21 shows a block diagram of the DM642 device in a video IP phone application.



**Figure 21. DM642 Based Video IP Phone**

From the software architecture standpoint, let us consider an IP phone based on H.323. H.323 is an umbrella of standards that specify the components, protocols and procedures that provide multimedia communication services—real-time audio, video, and data communications—over packet networks, including Internet protocol (IP)-based networks. H.323 includes standards that encompass call control, parameter negotiation, voice and video codecs (encoders and decoders). H.261 and H.263 are commonly used video compression algorithms. The video processing is typically the most compute intensive part of the phone. DM642 is capable of handling encode and decode of a single channel CIF (352x288) resolution video stream at high quality. The processor may also need to handle noise filtering and video pre processing functions. If the phone is designed to handle multiple video decodes, then additional scaling and video composition algorithms also need to be handled on the DSP.

Voice codecs commonly used in an IP phone include G.711 (standard PCM 64kbps), G.722 (wideband codec 48, 56 and 64kbps), G.723.1 (5.3 or 6.3kbps), G.726 (ADPCM at 16,24, 32 and 40kbps), G.728 (low delay CELP 16kbps) and G.729 (8kbps). Each of these codecs differs in complexity and performance. DM642 has the horsepower to run several channels of voice encode/decode on a single chip. Video and voice synchronization also needs to be performed on the processor. In an IP phone, echoes become more prominent since the aggregate network delay is large. The DSP also needs to handle line echo cancellation algorithms like G.168 that cancel echoes caused by signal reflection at the hybrid circuit in a telephone. Acoustic echo cancellation (needed for speakerphone feature support), voice activity detection (optimize network traffic), jitter buffer control etc are other functions typically implemented on the DSP. In a video IP phone solution the DM642 also handles all the functions required for establishing, maintaining and terminating a phone call including H.225, H.245 standards that are part of H.323. The DM642 has not only all the horse power to handle the video and voice signal processing it can also handle the network management functions of a video IP phone.

The DM642 device enables a low-cost full-featured video phone design with minimum glue and maximum flexibility.

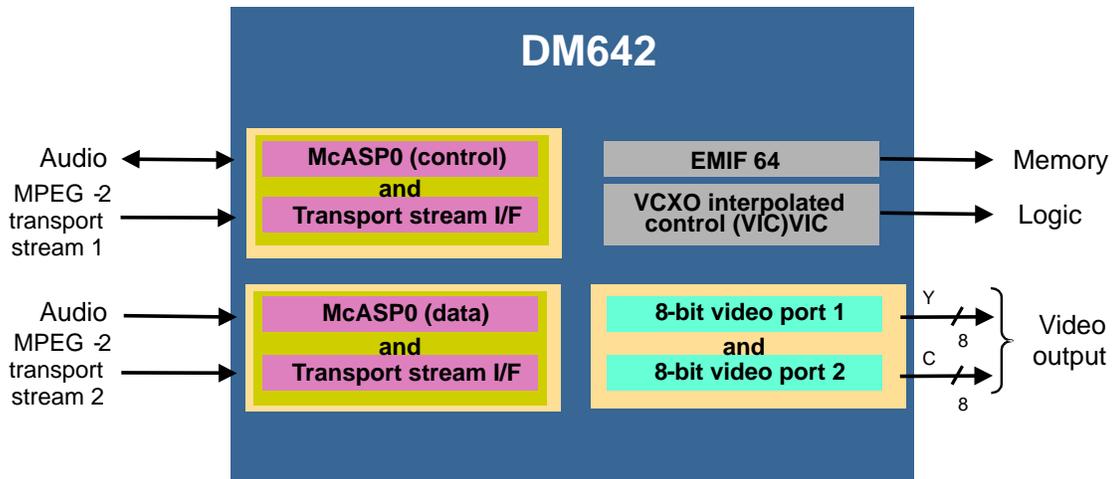
## Video on Demand Set-Top Box

Video on demand (VoD) set-top boxes enable users to select video content from a central server for viewing on a television or computer screen. VoD can be used for entertainment (ordering movies transmitted digitally), education (viewing training videos), and videoconferencing (enhancing presentations with video clips). VoD can be used to unleash services that offer, besides movies on demand, personalized news, sports packages as well as PVR (personal video recorder) services. These days, some set-top boxes double as a DVD player, web browser, and e-mail client as well. With the increasing deployment of broadband access services like high-speed cable modems and DSL, VoD services are becoming more popular and viable.

From a design standpoint, the set-top box needs to be able to handle the following functionality:

- **Content stream interface (satellite, terrestrial broadcast):** In the front end, the set-top box supports a QAM (cable), QPSK (satellite) or OFDM (terrestrial) demodulation unit. Set tops that support broadband may also include high-speed cable modem or DSL front end.
- **Video/audio decompression:** Video content is usually compressed for delivery due to the high bandwidth required for transmission. Set top boxes today typically provide MPEG2 decode capability to decompress the video content for viewing on the television monitor. With the penetration of broadband access services, there is a clear need for migration to video standards like MPEG4, H.26L and even proprietary standards that can provide high quality video at bit rates lower than 1Mbps. With a programmable processor based solution, deployment of video services based on different standards becomes possible quickly.
- **Middleware software:** Set-top boxes that provide internet access, video games and email capability support middleware software to handle email client and internet browsing capabilities. Services like electronic program guide for content selection and display are also typically supported.
- **User interaction:** Most set-top boxes support user interfaces like remote control, keypad or joysticks to select services.
- **Storage media interface:** Optionally, if the system is designed to support personal video recorder (PVR) functionality, then it needs to interface to storage media like hard disk drives.

The DM642 processor provides all the essential interfaces that make it perfect for use in a digital set top box application. Figure 22 illustrates a basic block diagram of a digital set-top box application using the DM642.



**Figure 22. Set-Top Box Application Using a DM642**

In a cable set-top box application, a Texas Instruments-enabled cable modem can be coupled with a DM64x to provide full STB functionality. The demodulated data from the modem is sent to the DM642 for MPEG2 video and audio decompression. The video ports on the DM642 have transport stream interface that interfaces to MPEG2 streams. Due to availability of multiple video ports, a DM642 based set top box can even support picture in picture functionality. At full speed, the DM642 can handle decode of 4 MPEG2 channels at SDTV resolution. The DM642 is capable of not only decoding an MPEG2 video stream but also of handling accompanying audio decode, network interface, and running all the required middleware software. The video ports on the DM642 provides direct interface to NTSC encoders that convert the uncompressed video to analog for television viewing. The NTSC encoders can be controlled by the I2C peripheral on the DSP. The multichannel audio serial port (McASP) is capable of interfacing directly to stereo audio streams (I2S). With the video port and McASP, the DM642 offers glueless connectivity to audio and video streams.

In a higher end set-top box system with added functionality, a micro controller may be used to support the control functions with the DSP being the primary video engine. In this application, the DSP can be seamlessly interfaced to the controller through the 66 MHz PCI bus. Besides handling control functions, the host processor can handle the video stream interface and send compressed data to the DSP through PCI if desired. The flexible peripherals on the DM642 allow it to be easily designed into either configuration of a set-top box application.

The advantage of using a programmable digital media processor in a set top application is that it enables software downloads for future upgrades. As the industry migrates to newer video standards, set-top boxes designed with a programmable solution can be quickly upgraded to handle the newer codecs.

## Surveillance Digital Video Recorder

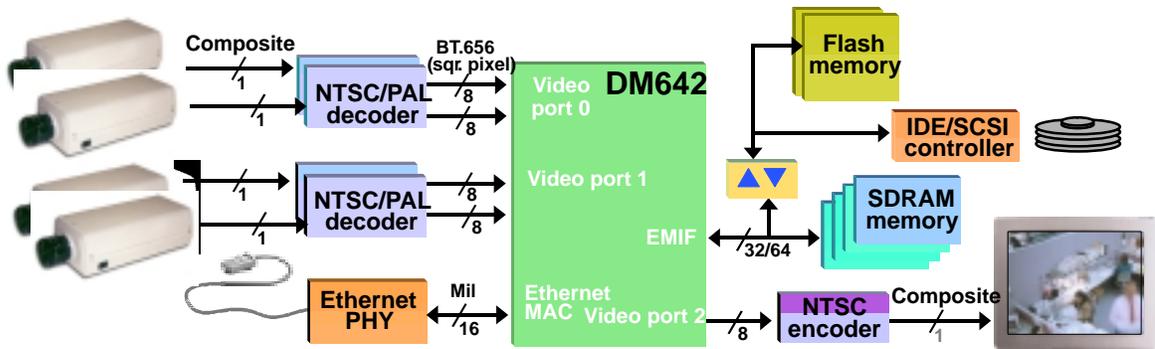
Video surveillance systems using closed-circuit television are increasing being deployed in locations around the world. These systems capture video from cameras and digitally compress it for storage and retrieval. Optionally, the compressed video can even be sent over to network to a remote location for archival or analysis. Video surveillance systems can be found in numerous locations including corporate facilities, financial institutions, retail stores, gas stations, casinos etc. These systems allow the area to be monitored from a central location while at the same time capturing the information so that it can be used as evidence if required.

Traditionally, a lot of the video security systems deployed have been analog whereby the data is captured and stored on videotapes. Digital video recording overcomes the limitations of traditional analog technologies and provides a more robust, reliable, efficient and high quality solution.

Digital solutions have the advantage that due to digitization of the signal, the system is no longer susceptible to wear and tear. Analog systems require a higher level of upkeep due to the mechanical moving parts in the system. A videotape head requires replacement about every 10,000 hours of operation and a cassette is usually deemed unusable after 2000 hours of operation. Digital systems on the other hand, retain high quality images irrespective of the storage media chosen. This becomes significant if the video is to be used as evidence in investigations. Digital systems using pixel based search technology also simplify the process of searching for a date or time stamp in the stored video down to a few seconds as compared to several minutes in the analog system. In addition, often this access and retrieval process can occur while simultaneously recording video. Availability of high-density hard disks has also increased the available capacity of storage. A 45GB hard disk can store millions of images as compared to a standard VHS tape storing 500,000 images. Digital technology also enables playback and storage of high quality images at 30fps and higher. Surveillance DVRs often trigger recording with alarm relays or motion detection circuits to conserve disk space and record only critical video footage. Digitally recorded images can also be printed without image quality loss.

DVR systems usually offer remote connectivity via TCP/IP, ISDN, cable, DSL etc. This allows users to dial in to the server and access the security camera while the system is still recording. This feature is used extensively to monitor parking lots, correction facilities, retail stores, warehouses etc. The networking capability is also used to transmit the compressed images to a remote location for storage. Surveillance DVR systems can also incorporate software biometrics like face recognition to limit user access to locations.

The DM642 has several features that lend themselves well to a video surveillance application. Figure 23 illustrates use of the DM642 in a DVR application.



**Figure 23. Networked DVR System Using a DM642**

The DM642 has video ports that enable glueless connectivity to multiple NTSC-compliant camera systems. Thus, the system can be scaled up to handle single or multiple camera inputs. The video port output can be used to send the data to a monitor for local viewing if desired. Typically, standards based DVR systems use JPEG, wavelet or H.263 compression. The DSP can be used to encode multiple streams of high quality video at different resolutions at any of the standards mentioned before. The 10/100Mbps ethernet MAC on the DSP can be used to send the compressed video over the network to a remote site (if required) thus enabling a networked DVR application. The DM642 has a flexible memory interface that can be used to connect the system to a hard disk drive for data storage. Advanced video surveillance systems come equipped with alarm relay systems that are used to trigger video recording. The flexible memory interface on the DM642 can be used to interface the DSP to alarm relay logic as well as a serial RS232 user interface. Security systems may also offer features like motion detection, camera pan/tilt/zoom, camera shake stabilization, video time stamping, etc. These functions can be easily implemented on the DSP with customized software packages. The processor-based system can also be used to host a web server to allow a remote access over the LAN or internet.

The advantage of a programmable solution is that the system can be quickly upgraded with value added features or can be adapted to adhere to emerging standards. The DM642 adds tremendous value in designing a low-cost, scalable, networked DVR.

## Ease of Development

The DM642 continues the C6000 DSP tradition of being a very friendly high-level language compiler target. The processor architecture and the compiler development continue to be closely coupled.

The C64x CPU of the DM642 continues the load/store architecture found in the C6000 family. By separating arithmetic and memory operations, processor throughput is maximized. The RISC like instruction set and extensive use of pipelining allow many instructions to be scheduled and executed in parallel. Also, because there is a great deal of orthogonality to the data path, register file, and instruction set, the compiler has very few restrictions. For example, the general-purpose registers can be used for data or data address pointers. The ADD instruction can execute on six of the eight functional units giving the compiler many choices of where to execute the ADD.

Other improvements to the C64x architecture, which increase compiler performance, are being able to execute logical instructions on two additional functional units, large number of general-purpose registers and condition registers. In addition, instructions have been added that further reduce code size and increase register flexibility.

TI's C6000 compile tools were co-developed with the architecture to offer best-in-class performance. Compiler performance will be further enhanced as the advantages of the C64x VelociTI.2 extensions are more fully leveraged in subsequent compiler releases. For further details on specific optimization techniques for the C64x, please refer to the C6000 Programmer's Guide. For more information on compiler optimization, please see the C6000 Compiler Optimization Tutorial at the URL <http://www.ti.com> as well.

## Summary

The DM642 addresses the needs of video and imaging application developers. The heart of the DM642 is the C64x CPU which includes special instructions to accelerate the performance of video and image processing. Also, the RISC-like instruction set and extensive use of pipelining in C64x, allow many instructions to be scheduled and executed in parallel. Parallelism is the key to extremely high performance needed for video and imaging applications. A high performance two-level cache design allows the CPU to operate at the maximum rate. The two-level cache lowers development time by automating off-chip to on-chip data transfers. A high performance EDMA controller feeds the CPU through a flexible high-bandwidth bus architecture.

The DM642 includes a rich suite of peripherals targeted for video and imaging applications. These peripherals allow system cost reduction and faster system development. Peripherals are included that support glueless interfaces for BT.656 video I/O, HDTV I/O at upto 10-bits per component, RGB I/O, MPEG-2 transport stream interfaces, ethernet MAC support, and audio interfaces for I2S, DIT, etc.

## Appendix A Comparison of DM642 ot C64x DSPs

A comparison of DM642 to C64x DSPs is shown in Table A-1. For further details please see: <http://www.ti.com>

**Table A-1. Comparison of DM642 to C64x DSPs**

	<b>C6411</b>	<b>C6414</b>	<b>C6415</b>	<b>C6416</b>	<b>DM642</b>
<b>Performance</b>	2400 MIPS	4000/4800 MIPS	4000/4800 MIPS	4000/4800 MIPS	4000/4800 MIPS
<b>Memory (Bytes)</b>	L1 Prog: 16K L1 Data: 16K L2 P/D: 256K	L1 Prog: 16K L1 Data: 16K L2 P/D: 1024K	L1 Prog: 16K L1 Data: 16K L2 P/D: 1024K	L1 Prog: 16K L1 Data: 16K L2 P/D: 1024K	L1 Prog: 16K L1 Data: 16K L2 P/D: 256K
<b>Key Features</b>	HPI or 33 MHz PCI	HPI	33 MHz PCI, Utopia 2	VCP/TCP accelerators	3 Videoports 66 MHz PCI EMAC
<b>BGA Package</b>	23 mm/532	23 mm/532	23 mm/532	23 mm/532	23 mm/548
<b>Internal Power (Typical)</b>	0.25 W @ 300 MHz <sup>†</sup>	0.64/1.04 W @ 500/600 MHz <sup>†</sup>	0.64/1.04 W @ 500/600 MHz <sup>†</sup>	0.64/1.04 W @ 500/600 MHz <sup>†</sup>	0.64/1.04 W @ 500/600 MHz <sup>†</sup>

<sup>†</sup> Estimated Power

## Appendix B Related Documents

A great deal of literature is available today for the C6000 devices. Some documents are listed below. These and other related documents may be downloaded from: <http://dspvillage.ti.com>

- TMS320C6000 CPU and Instruction Set Reference Guide (SPRU189)
- Manual Update Sheet for TMS320C6000 CPU and Instruction Set Reference Guide (SPRZ168)
- TMS320C6000 Peripherals Reference Guide (SPRU190)
- TMS320C6000 Technical Brief (SPRU197)
- TMS320C64x Technical Overview (SPRU395)
- Code Composer Studio User's Guide (SPRU328)
- TMS320C6000 Code Composer Studio Tutorial (SPRU301)
- TMS320C6000 Programmer's Guide (SPRU198)
- TMS320C6000 Evaluation Module User's Guide (SPRU269)
- TMS320C6000 Chip Support Library API Reference Guide (SPRU401)
- TMS3206000 Peripheral Support Library Programmer's Reference (SPRU273)
- TMS320C6000 Assembly Language Tools User's Guide (SPRU186)
- TMS320C6000 Optimizing C Compiler User's Guide (SPRU187)
- TMS320C6000 C Source Debugger User's Guide (SPRU188)
- TMS320C6000 C Source Debugger For SPARCstations (SPRU224)
- TMS320C6000 DSP/BIOS User's Guide (SPRU303)
- TMS320C6000 DSP/BIOS Application Programming Interface (API) Reference Guide (SPRU403)
- DSP Algorithm Integration Standard (XDAIS) Rules and Guidelines (SPRU352)
- DSP Glossary (SPRU258)
- TMS320 DSP Standard Algorithm Developer's Guide (SPRU424)

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

### Mailing Address:

Texas Instruments  
Post Office Box 655303  
Dallas, Texas 75265