# Chapter 3 Environments of DSP Implementation

## 3.1 The DSP Board

The DSP board used in our implementation is the VP[3] (Video Parallel Programmable Processing platform) board, which is a powerful platform dedicated to video applications. Fig 3-1 shows the block diagram of VP[3]. It is a PCI compatible DSP card with eight TMS320DM642 DSPs runnig at 600MHz. VP[3]'s features are listed as below.

1. Eight TMS320DM642 DSPs run at up to 600MHZ (Fixed-point).
2. Each DSP has a private memory of 128MB which is SDRAM running at 100MHz with 64-bit bandwidth.
3. Each DSP has three powerful configurable video ports which are used as communication ways. That can make the eight processors to be arranged as a simple pipeline, with a fully parallel scheme.
4. DSPs can communicate with each others in the following ways:
   I. Direct memory-to-memory block exchanges through the DMA controller services.
   II. Video data or raw data through their video ports.
   III. The host buses of the DSPs are linked to a PCI interface and can send/receive messages to/from the host through the PCI interface.
5. VP3 includes a main board and daughter board. The main includes a PCI interface and the daughter board includes the video and audio I/O.

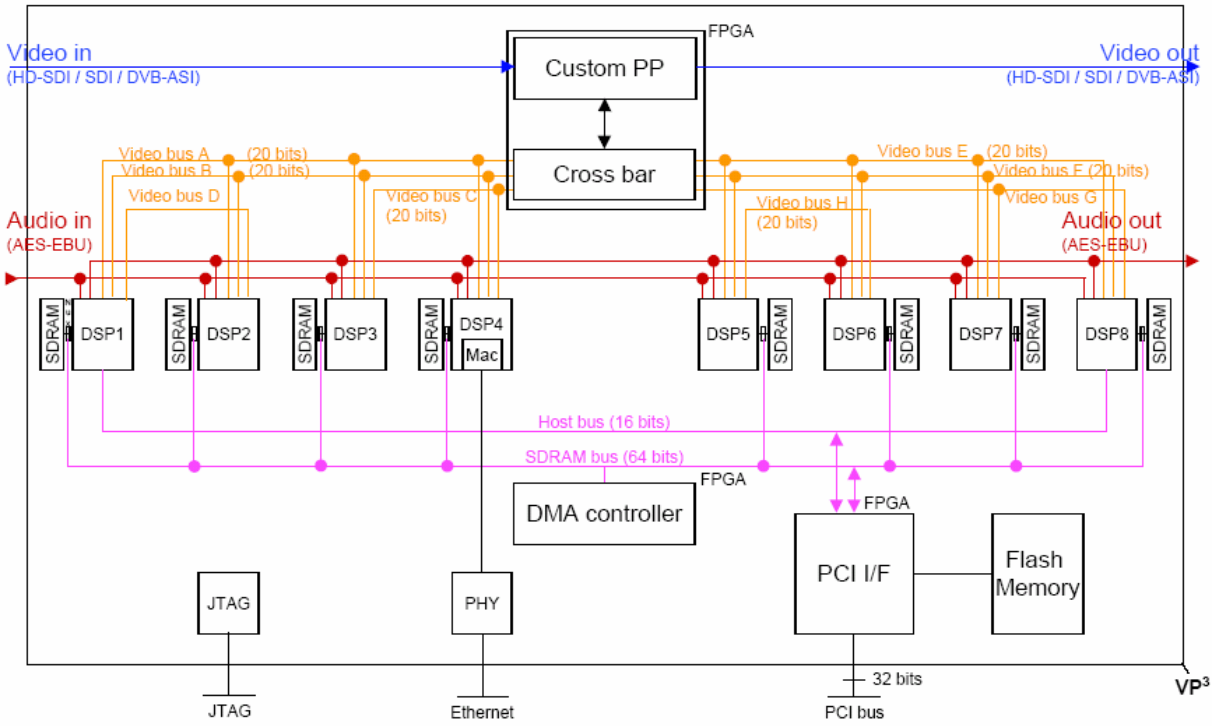


Fig 3-1 Block diagram of VP$^3$ [6]

# 3.2 DSP Chip

The DSP chip used in our implementation is TI's TMS320DM642. It is based on the second-generation high-performance, advanced VelociTI™ very long instruction word (VLIW) architecture (VelociTI.2™) developed by Texas Instruments (TI), The VLIW architecture makes these DSPs an excellent choice for digital media applications. VLIW achieves high performance through instruction-level parallelism , performing multiple instruction during a single cycle. VelociTI is a highly deterministic architecture, having few restrictions on how or when to fetch, execute, and store instructions. VelociTI.2 is an extension of VelociTI, with the following advanced feature.

- Eight highly independent functional units with VelociTI.2 extensions:
  - Six ALUs (32-/40-Bit), each supports single 32-Bit, dual 16-Bit, or quad 8-Bit arithmetic per clock cycle
  - Two multipliers support four 16 × 16-bit multiplies (32-bit results) per clock cycle or eight 8 × 8-bit multiplies (16-bit Results) per clock cycle

- Load store architecture with non-aligned support
- 64 32-bit general purpose registers
- Reduction of code size by instruction packing.
- Conditional instruction

The DM642 DSP is a Video/Imaging fixed-point digital signal processor in the TMS320C64x family. It has eight independent functional units running at 600MHZ for peak execution of4800 MIPS . Some key feature of DM642 are listed below.

- Eight highly independent functional units—two multipliers to generate 32-bit result and six arithmetic logic units (ALUs)
- The VelociTI.2™ extensions in the eight functional units include new instructions to accelerate the performance in video and imaging manipulations and to extend the parallelism of the VelociTI™ architecture.
- Conditional execution reduces cost of branch and increase parallelism.
- Instruction packing reduces code size, program fetches, and power consumption.
- 8/16/32/40-bit data support.
- Saturation and normalization provide support for key arithmetic operations.

Fig 3-2 shows the block diagram of the TMS320DM642. The DSP can be divided into two parts: central processing unit (CPU) and memory architecture. We will describe each part in the following subsections.

### 3.2.1 Central Processing Unit (CPU)

The C64x central processing units shown in Fig 3.2 include the following parts:

- Program fetch unit
- Instruction dispatch unit and advanced instruction packet
- Instruction decode unit
- Two data paths, with four functional units in each path
- 64 32-bit registers

- Control registers
- Control logic
- Test, emulation, and interrupt logic

Program fetch, instruction dispatch, and instruction decode units can deliver up to eight 32-bit instructions to the functional units during every CPU clock cycle. The processing of instructions occurs in each of the two data paths (A and B). Each path contains four functional units (.L, .S, .D and .M) and 32 32-bit general-purpose registers. The function units are described in Table 3.1 and Table 3.2.

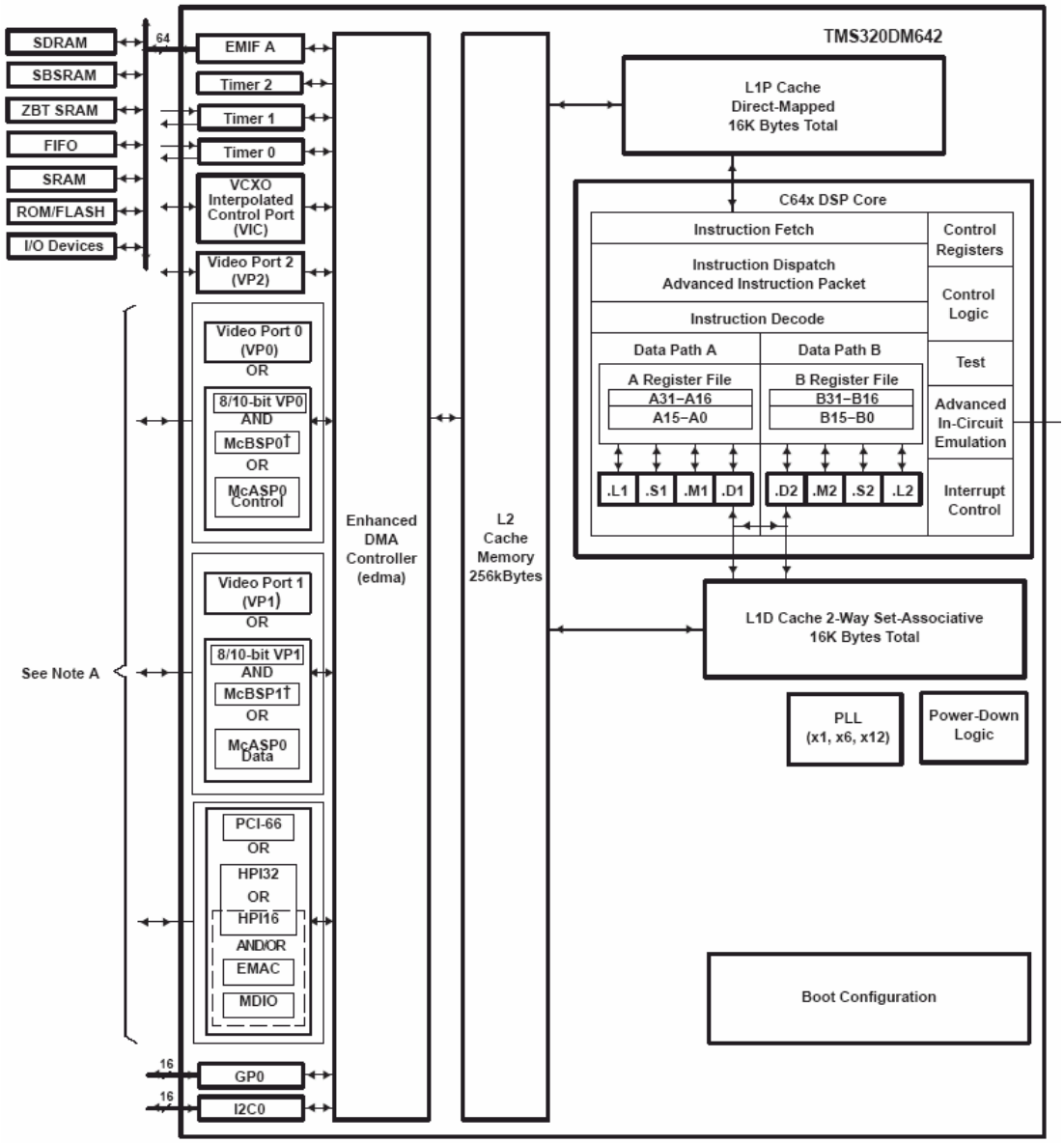


Fig 3-2 Block diagram of TMS320DM642 [7]

There are two register files (A and B) in the C6000 data paths. The number of general-purpose register in C64x is twice of that in C62x/C67x. The register files are connected to the opposite side functional units via 1X and 2X cross paths which allow functional units from one data path to access a 32-bit operand from an opposite side register file.

Each data path has 8 functional units for multiplication operations (.M), logical and, arithmetic operations (.L), branch, bit manipulation, and arithmetic operations (.S), andloading/storing and arithmetic operations (.D). Table 3-1 and Table 3-2 show these functional units and their operations.

Most data lines in the CPU support 32-bit operands, while some support long(40-bit) operands. Each functional unit has its own 32-bit write port to a general-purpose register file and 32-bit read ports for source operands src1 and src2 (refer to Fig 3-3). Four units (.L1, .L2, .S1, .S2) have an extra 8-bit-wide port for 40-bit long writes and reads.

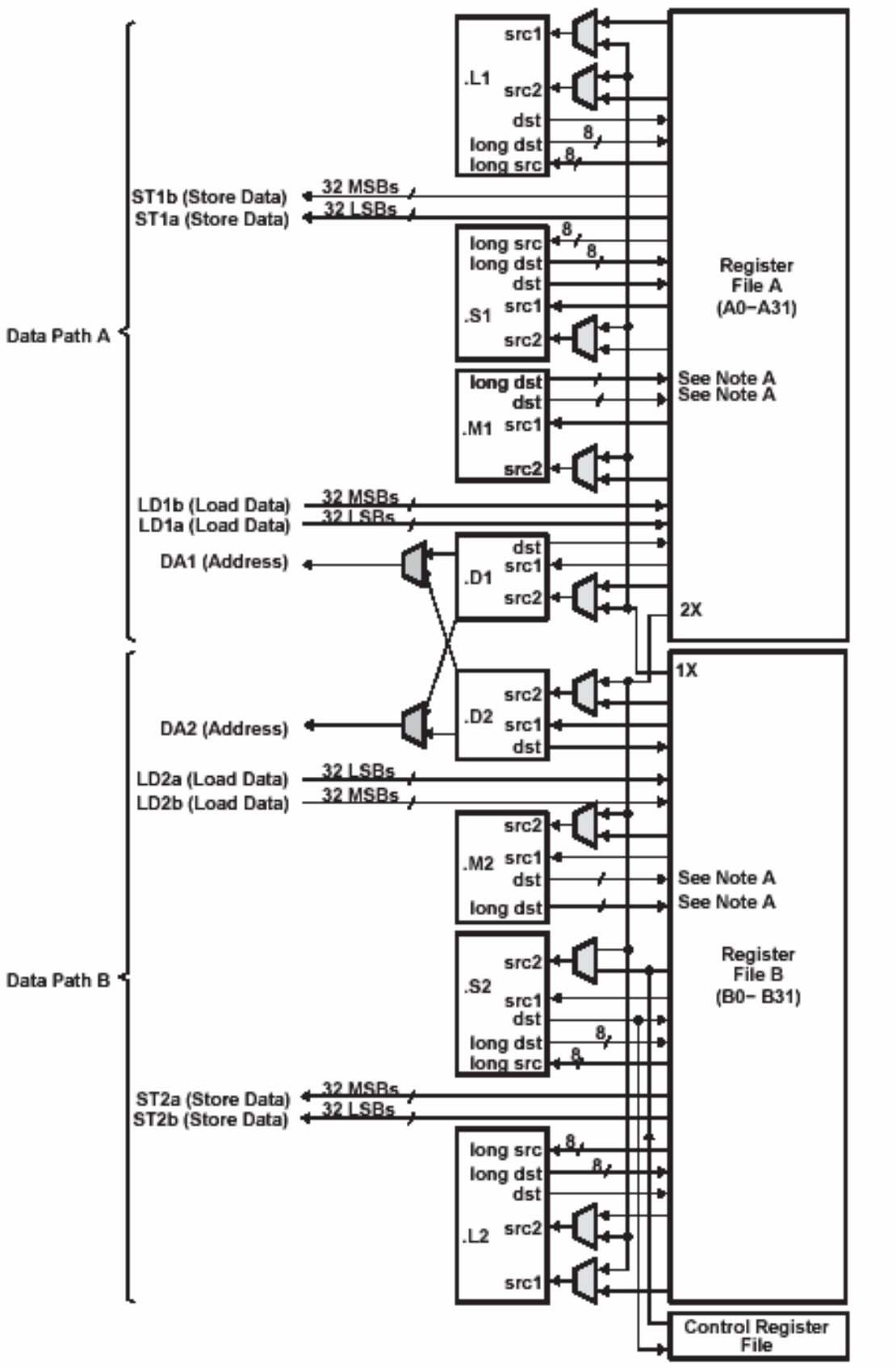


Fig 3-3 Data Path [7]

Table 3-1 Functional Units and Operations Performed [8]

| Functional Unit | Fixed-Point Operations | Floating-Point Operations |
|---|---|---|
| .L unit (.L1, .L2) | 32/40-bit arithmetic and compare operations<br>32-bit logical operations<br>Leftmost 1 or 0 counting for 32 bits<br>Normalization count for 32 and 40 bits<br>**Byte shifts**<br>**Data packing/unpacking**<br>**5-bit constant generation**<br>**Dual 16-bit arithmetic operations**<br>**Quad 8-bit arithmetic operations**<br>**Dual 16-bit min/max operations**<br>**Quad 8-bit min/max operations** | Arithmetic operations<br>DP → SP, INT → DP, INT → SP conversion operations |
| .S unit (.S1, .S2) | 32-bit arithmetic operations<br>32/40-bit shifts and 32-bit bit-field operations<br>32-bit logical operations<br>Branches<br>Constant generation<br>Register transfers to/from control register file (.S2 only)<br>**Byte shifts**<br>**Data packing/unpacking**<br>**Dual 16-bit compare operations**<br>**Quad 8-bit compare operations**<br>**Dual 16-bit shift operations**<br>**Dual 16-bit saturated arithmetic operations**<br>**Quad 8-bit saturated arithmetic operations** | Compare<br>Reciprocal and reciprocal square-root operations<br>Absolute value operations<br>SP → DP conversion operations |

Table 3-2 Functional Units and Operations Performed [8]

| Functional Unit | Fixed-Point Operations | Floating-Point Operations |
|---|---|---|
| .M unit (.M1, .M2) | 16 x 16 multiply operations<br>**16 x 32 multiply operations**<br>**Quad 8 x 8 multiply operations**<br>**Dual 16 x 16 multiply operations**<br>**Dual 16 x 16 multiply with add/subtract operations**<br>**Quad 8 x 8 multiply with add operation**<br>**Bit expansion**<br>**Bit interleaving/de-interleaving**<br>**Variable shift operations**<br>**Rotation**<br>**Galois Field Multiply** | 32 X 32-bit fixed-point multiply operations<br>Floating-point multiply operations |
| .D unit (.D1, .D2) | 32-bit add, subtract, linear and circular address calculation<br>Loads and stores with 5-bit constant offset<br>Loads and stores with 15-bit constant offset (.D2 only)<br>**Load and store double words with 5-bit constant**<br>**Load and store non-aligned words and double words**<br>**5-bit constant generation**<br>**32-bit logical operations** | Load doubleword with 5-bit constant offset |

### 3.2.2 Memory Architecture

The C64x memory architecture consists of the following parts:

1. Internal data/program memory.
2. External memory interface (EMIF)
3. Enhanced Directed-Memory-Access (EDMA)
4. Host-port interface (HPI)
5. Two Multichannel-buffered serial ports (McBSPs)
6. Three Configurable Video Ports
7. Interrupt selector
8. Three 32-bit general purpose timers
9. Multichannel Audio Serial Port (McASP)

The DM642 uses a two-level cache-based architecture and has a powerful set of peripherals. The Level 1 program cache (L1P) is a 128-Kbit direct mapped cache and the Level 1 data cache (L1D) is a 128-Kbit 2-way set-associative cache. The Level 2 memory/cache (L2) consists of a 2-Mbit memory space that is shared by both program space and data space. The TMS320DM642 internal program memory can be mapped into the CPU address space or operated as a program cache. There is a single port to access internal program memory, with an instruction fetch width of 256 bits. The internal data memory on C64x devices divides the memory into eight 32-bit wide banks. These banks are single-ported, allowing only one access per cycle. This is in contrast to the C621x/C671x devices, which use a single bank of dual-ported memory rather than multiple banks of single-ported memory. There are more details described in [9].

The EMIF provides the interfaces for the DSP core and the external memory. It includes:

- Pipelined synchronous-burst SRAM (SBSRAM)
- Synchronous DRAM (SDRAM)
- Asynchronous devices, including SRAM, ROM, and FIFOs

- An external shared-memory device

The C64x DSP has two EMIFs: EMIFA and EMIFB. DM642 only has EMIFA which can provide up to 64-bit wide capability to all types of memory.

The enhanced direct memory access (EDMA) controller handles all data transfers between the Level-two (L2) cache/memory controller and the device peripherals on the C64x DSP. The EDMA controller in the C64x DSP has a different architecture from the previous DMA controller in the C620x/C670x devices. The EDMA includes several enhancements to the DMA, such as 64 channels for the C64x DSP, with programmable priority, and the ability to link and chain data transfers. The EDMA allows movement of data to/from any addressable memory spaces, including internal memory, peripherals, and external memory.

Table 3-3 Difference between C62x/C67x and C64x HPI [10]

| | C62x/C67x HPI | | C64x HPI | |
|---|---|---|---|---|
| **Features** | **C620x/C670x** | **C621x/C671x** | **HPI16** | **HPI32** |
| Data bus width | 16-bit | 16-bit | 16-bit | 32-bit |
| Byte enable $\overline{\text{HBE}}$[1–0] pins | Yes | No | No | No |
| HHWIL | Used | Used | Used | Not used |
| Single halfword access support | Yes | No | No | No |
| HPIA access | By host only | By host only | By host or CPU. HPIA consists of HPIAR and HPIAW. | By host or CPU. HPIA consists of HPIAR and HPIAW. |
| $\overline{\text{HRDY}}$ operation | Not-ready after each word access | Not-ready only if internal read/write buffers not ready | Same as C621x/C671x HPI | Same as C621x/C671x HPI |
| Internal read buffer | No | Yes, 8-deep | Yes, 16-deep | Yes, 16-deep |
| Internal write buffer | No | Yes, 8-deep | Yes, 32-deep. Flushes after internal timer times out | Yes, 32-deep. Flushes after internal timer times out |

The host port interface (HPI) is a parallel port through which a host processor can directly access the CPU memory space. The host device functions as a master to the interface, which increases the ease of access. Both the host and the CPU can access the HPI control register (HPIC) and the HPI address register (HPIA). The host can access the HPI data register (HPID) and the HPIC by using the external data and interface control signals. Table 3-3 shows the difference between C62x/C67x and

C64x. There are more details described in [10].

The McBSP consists of a data path and a control path that connect to external devices. Separate pins for transmission and reception communicate data to these external devices. The device communicates to the McBSP using 32-bit-wide control registers accessible via the internal peripheral bus. There are more details described in [11].

The DM642 device has three configurable video port peripherals. These video port peripherals provide an interface to common video decoder and encoder devices. The DM642 video port peripherals support multiple resolutions and video standards. These three video port peripherals are configurable and can support video capture and/or video display modes. As shown in Fig 3-4, each video port consists of two channels — A and B with a 5120-byte capture/display buffer being splittable between these two channels. The video port peripheral can operate as a video capture port, a video display port, or a transport stream interface (TSI) capture port. For the capture mode, the video port may operate as two 8/10 bits channels of BT.656 or raw video. It may also operate 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 the display mode, the video port may operate 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. It may also operate in a two-channel 8/10-bit raw mode. There are more details described in [12].

The C64x has three 32-bits general-purpose timers that can be used to time event, count events, generate pulses, interrupt the CPU, and send synchronization events to the DMA. There are more details described in [13].

The other parts of peripherals are too low-level for user applications. For more details, refer to [14][15].

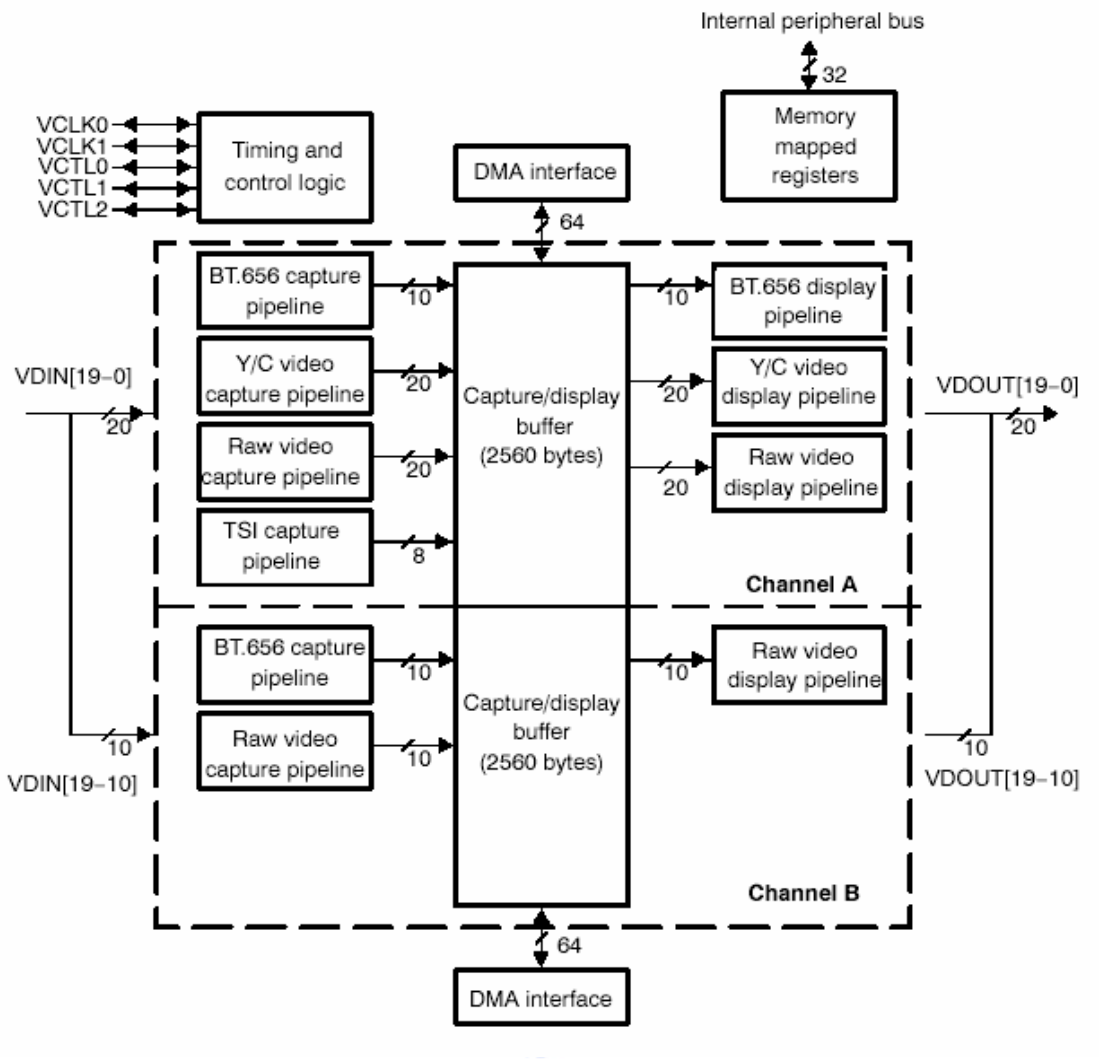


Fig 3-4 Video Port Block Diagram [12]