Hardware Architecture Overview

Publication date: October 2001

Sony Computer Entertainment Inc.
1-1, Akasaka 7-chome, Minato-ku
Tokyo 107-0052, Japan

Sony Computer Entertainment America
919 E. Hillsdale Blvd., 2nd floor
Foster City, CA 94404

Sony Computer Entertainment Europe
30 Golden Square
London W1F 9LD, U.K.

Table of Contents

About This Manual

This manual is the Runtime Library Release 2.4 version of the *Hardware Architecture Overview*.

It describes the PlayStation 2 system architecture and provides dataflow information.

Changes Since Last Release

None

Related Documentation

The “Software Architecture Overview” (SysSoft) document provides information on software structure, kernel, libraries, etc.

Note: the Developer Support Web site posts current developments regarding the Libraries and also provides notice of future documentation releases and upgrades.

Typographic Conventions

Certain Typographic Conventions are used throughout this manual to clarify the meaning of the text:

Convention	Meaning
`courier`	Indicates literal program code.
italic	Indicates names of arguments and structure members (in structure/function definitions only).
medium bold	Indicates data types and structure/function names (in structure/function definitions only).
blue	Indicates a hyperlink.

Developer Support

Sony Computer Entertainment America (SCEA)

SCEA developer support is available to licensees in North America only. You may obtain developer support or additional copies of this documentation by contacting the following addresses:

Order Information	Developer Support
In North America:	*In North America:*
Attn: Developer Tools Coordinator Sony Computer Entertainment America 919 East Hillsdale Blvd. Foster City, CA 94404, U.S.A. Tel: (650) 655-8000	E-mail: PS2_Support@playstation.sony.com Web: http://www.devnet.scea.com/ Developer Support Hotline: (650) 655-5566 (Call Monday through Friday, 8 a.m. to 5 p.m., PST/PDT)

Sony Computer Entertainment Europe (SCEE)

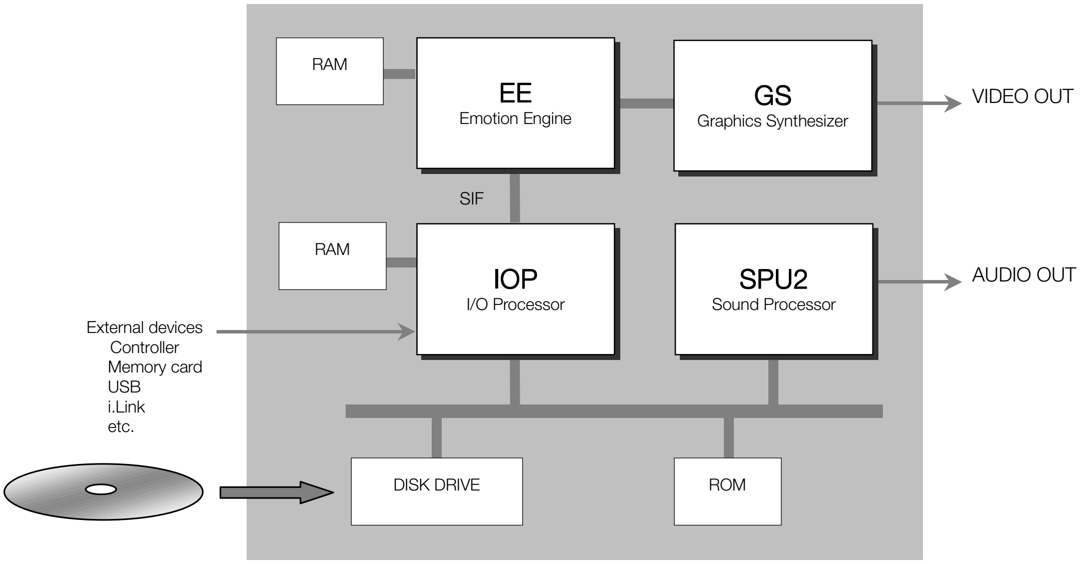
SCEE developer support is available to licensees in Europe only. You may obtain developer support or additional copies of this documentation by contacting the following addresses:

Order Information	Developer Support
In Europe:	*In Europe:*
Attn: Production Coordinator Sony Computer Entertainment Europe 30 Golden Square London W1F 9LD, U.K. Tel: +44 (0) 20 7859-5000	E-mail: ps2_support@scee.net Web: https://www.ps2-pro.com/ Developer Support Hotline: +44 (0) 20 7859-5777 (Call Monday through Friday, 9 a.m. to 6 p.m., GMT)

Architecture

The following is a block diagram of the overall PlayStation 2 system architecture.

Figure 1: PlayStation 2 block diagram

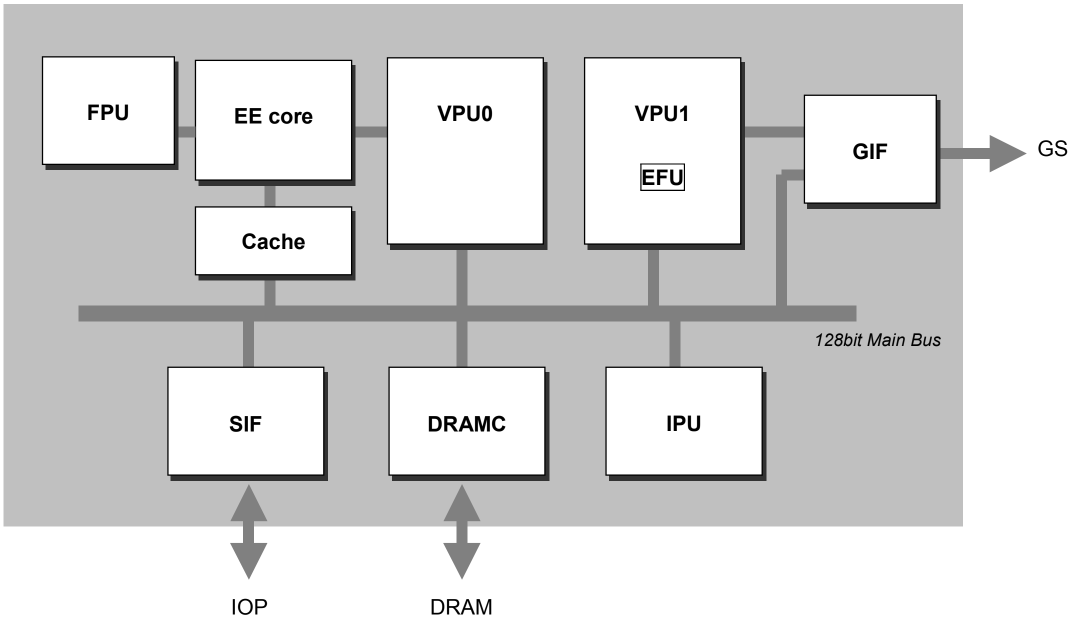


The main processor of the PlayStation 2 is known as the Emotion Engine (EE). The EE sends polygon data to the Graphics Synthesizer (GS) which performs drawing and outputs the result as a video signal. The IOP is a subprocessor that controls external input/output devices. The IOP has its own memory and can perform certain operations independently. The SPU2 provides functions related to sound.

Emotion Engine (EE)

The Emotion Engine (EE) is a processor designed specifically for computer entertainment applications that can perform high-speed, highly parallel operations. A block diagram of the Emotion Engine is shown below.

Figure 2: Emotion Engine Block Diagram



The EE core is a 64-bit microprocessor with a 2-way superscalar pipeline that can execute two instructions simultaneously. The instruction set is based on the MIPS III instruction set with certain MIPS IV and 128-bit multimedia extensions. The EE has a 16 KByte instruction cache, an 8 KByte data cache and in addition to an Uncached mode that bypasses the data cache, an Uncached Accelerated mode that uses a buffer to continuously accelerate access to adjacent memory addresses allowing memory accesses to be precisely controlled.

The EE also has 16 Kbytes of scratchpad memory (SPRAM) that functions as a high-speed work memory. The scratchpad memory can be accessed in the same manner as ordinary memory and can also be accessed through DMA data transfers between main memory or peripheral devices. Using the scratchpad memory as a double buffer makes it possible to send calculation results in the EE core to peripheral devices without interruption.

Vector Processing Unit (VPU)

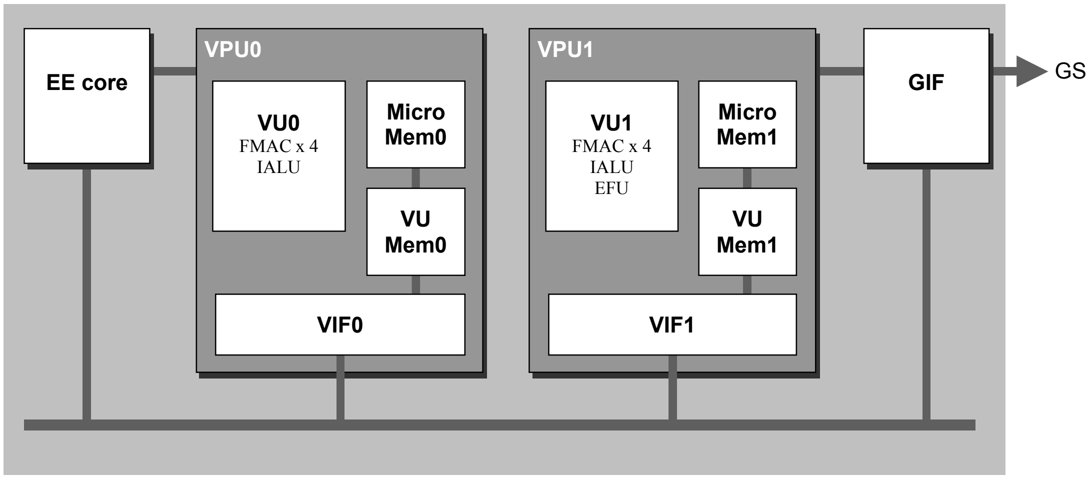
The VPU is a 32-bit floating-point vector processor that can perform four parallel operations. This architecture provides efficient processing of 4- element vectors and 4x4 matrices, which are often used in geometry calculations for 3D graphics.

The VPU has its own internal program memory and data memory and can execute programs using its own instruction set (micro-instruction). The micro-instruction field is 64 bits wide and can simultaneously execute an Upper Instruction, such as a 4-way parallel floating-point calculation, and a Lower Instruction, such as divide, load, store, integer arithmetic or branch.

The EE is equipped with two of these VPUs. VPU0 is connected to the EE core as a coprocessor. VPU1 operates independently and contains an elementary function unit (EFU) as well as expanded internal memory. It is connected to the GS through the GIF.

Work can be distributed either serially or in parallel across the functional units. For example, a serial distribution of work might have the EE core+VPU0 performing game logic calculations after which VPU1 would perform polygon data transformations. Alternatively, a parallel distribution might have the EE core+VPU0 performing complex, dynamic operations such as character animation while at the same time, VPU1 would be performing static operations such as background processing.

Figure 3: VPU Block Diagram



VIF

The VIF is an interface unit that connects the main bus of the EE to the VPU. The VIF provides a feature that allows data which is transferred to the VPU to be formatted so that it can be easily used by the VPU and written to the VPU's data memory. In addition, the VIF's internal registers allow static data to be generated in memory without having to transfer data each time.

VIF1, which is connected to VPU1, supports double buffering. Two buffers are provided in data memory to increase throughput. Double buffering allows VPU1 to be processing data while simultaneously transferring additional data to the other buffer.

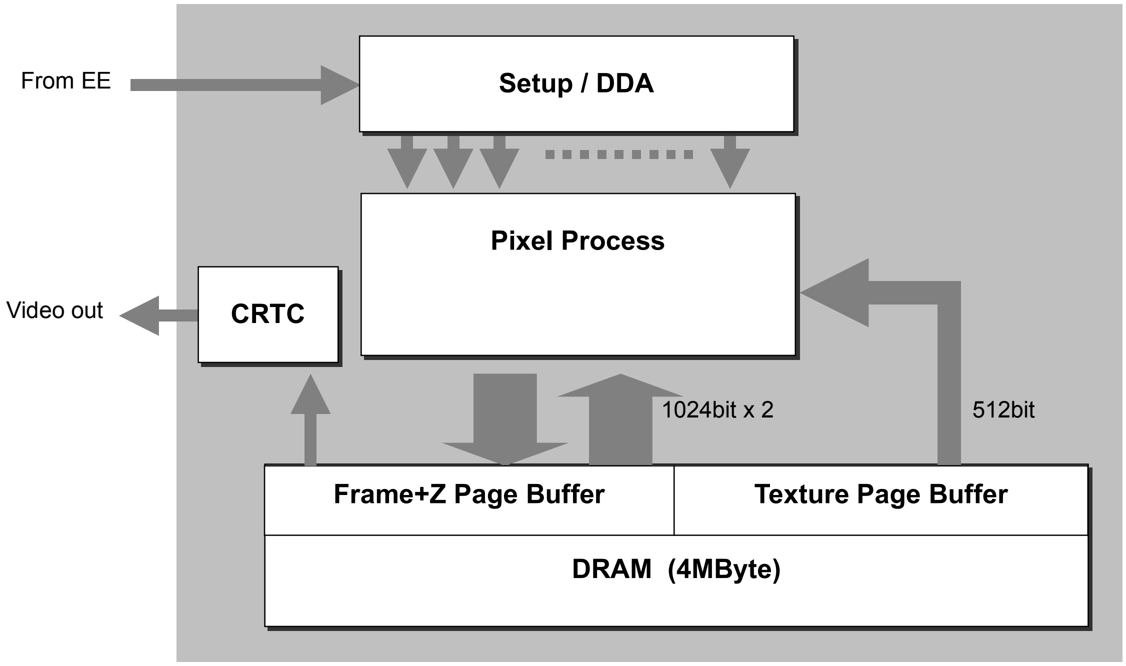
Image Processing Unit (IPU)

The IPU is an image data processor that decodes MPEG2 bitstreams. The IPU performs functions such as macroblock decoding, RGB conversion, automatic alpha plane generation, ordered dithering and 4-bit indexed color conversion. Motion compensation (MC) is performed using the multimedia instructions in the CPU core rather than through the IPU.

Graphics Synthesizer (GS)

The Graphics Synthesizer (GS) is a graphics processor that performs 3D graphics drawing as well as video signal output. The GS provides superior performance through the use of parallel processing and high-bandwidth memory access. A block diagram of the Graphics Synthesizer is shown below.

Figure 4: Graphics Synthesizer Block Diagram



The Setup/DDA unit performs pixel expansion on polygon data (GS primitives) that it receives from the Emotion Engine. Then the Pixel Processing unit performs texture mapping, fogging, and alpha blending for each pixel. A maximum of 16 pixels can be processed in parallel.

In this series of drawing operations, semi-constant information (drawing environment) such as texture and drawing mode are used in addition to polygon information. Switching can be performed with no overhead by using two sets of registers as part of the drawing environment. This allows the GS primitives of foreground objects and background objects to be sent out of order to the GS without a reduction in drawing speed.

Internally, 4 MBytes of DRAM are available. Frame buffers, Z-buffers, texture buffers, and CLUTs can be arbitrarily placed in memory, as desired.

The CRTC has two independent rectangle reading circuits. Separate regions of the frame buffer are read and different settings are made for image format, output resolution, and screen position. The results are then superimposed and output. This makes it easy to create a composite display, such as a 3D graphic animation and a message composed of high-resolution characters.

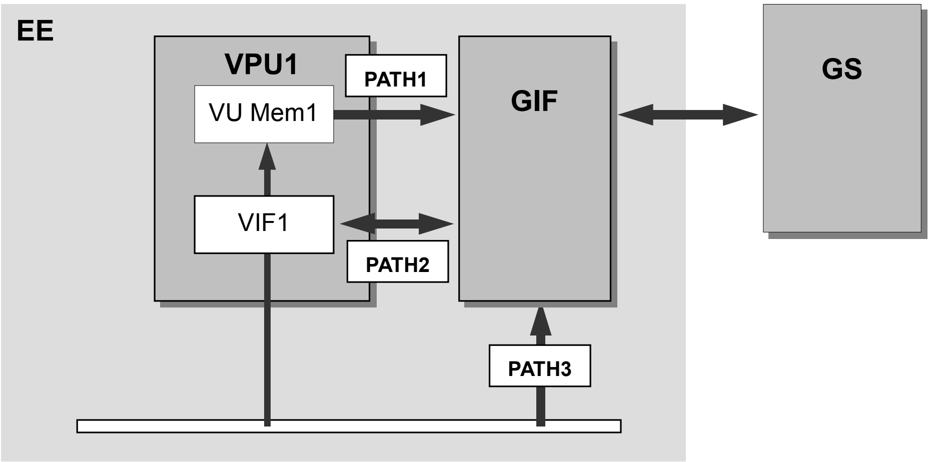
GIF

The GIF is the EE-to-GS interface unit. The GIF performs arbitration between three datapaths (described below). The GIF also has a feature to write data such as polygon data and image data to the appropriate GS registers. This operation is based on tags (GIFtags) which are added to the start of the data packets.

There are three datapaths from the EE to the GS: PATH1, PATH2, and PATH3. They rank in priority as PATH1>PATH2>PATH3, and they are used as follows:

- PATH1: Data transfer path from VPU1 data memory to the GS. Used to transfer GS primitive data generated by VPU1.
- PATH2: Data transfer path from VIF1 to the GS and from the GS to VIF1.
- PATH3: Data transfer path from EE main memory or scratchpad memory to the GS. To prevent blocking other drawing operations, large image data can be divided up and transferred separately. Transfers can also be blocked when necessary.

Figure 5: PATH1 / PATH2 / PATH3

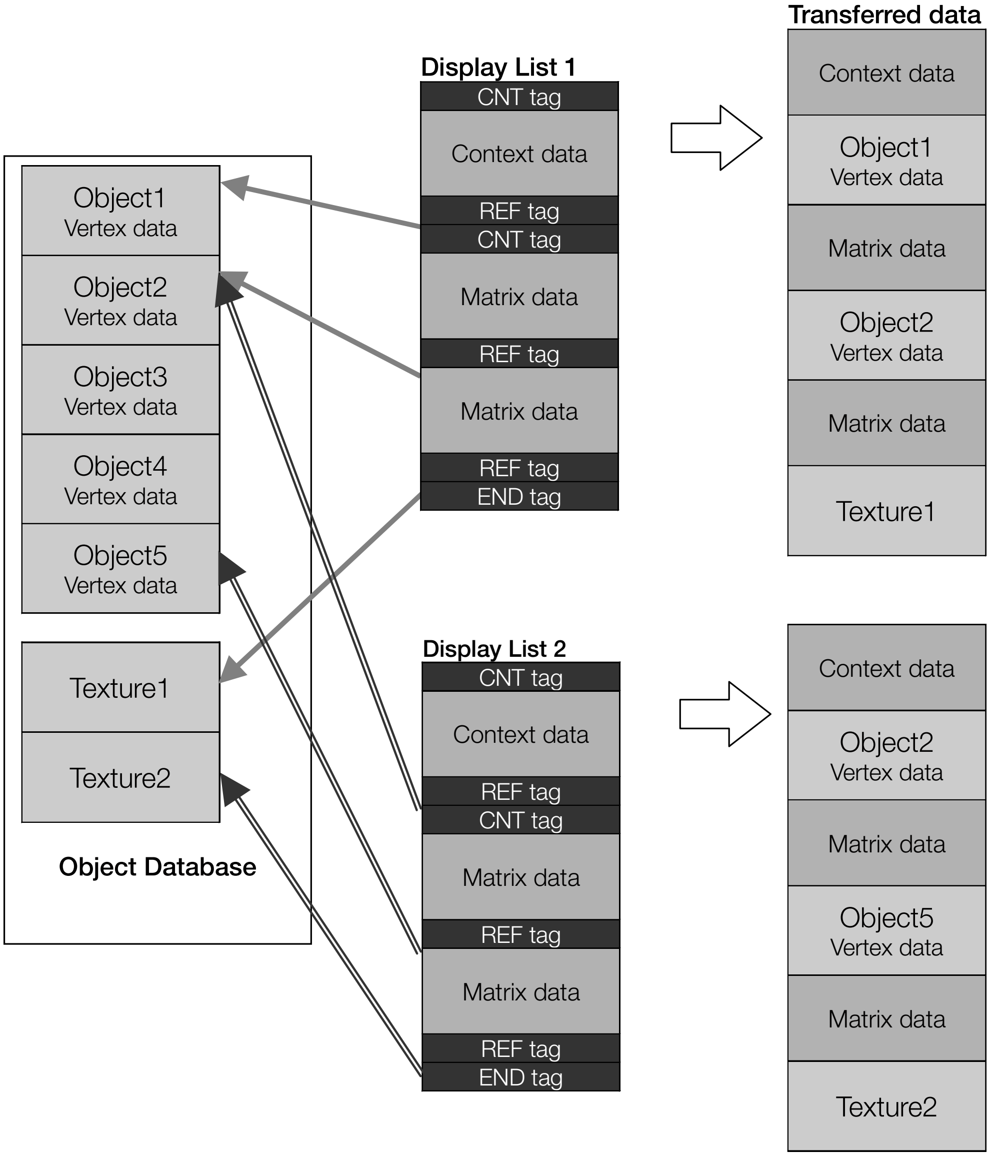


DMAC

Expanded DMAC features are provided so that the EE core's performance is not degraded by data transfer operations.

An important feature of the DMAC is the active DMA transfer (Chain Mode), which is performed using tags. Tags contain the address and size of the data to be transferred as well as the address of the tag that is next in the chain. The DMAC interprets these tags when it transfers data. For example, if a shared object database is being used, graphics data which is locally different can be generated by changing only the tag information.

Figure 6: DMA Transfer Using Tags



Other features include the stall control feature, which allows smooth peripheral-to-peripheral data transfers via memory, the priority control feature, which prioritizes specific data during transfers, and interleave mode, which can be used to crop image data.

IOP

The IOP is a processor that controls external input/output devices such as the CD/DVD-ROM drive, the sound device (SPU2), controllers, PS2 memory cards, PCMCIA, USB, and i.Link. The IOP provides processing features equivalent to those of the PlayStation CPU and can work together with the EE via the SIF while performing independent processing.

The SIF consists of three DMA channels and signal lines attached to each channel. Mutual data transfers occur between EE memory and IOP memory allowing interrupts to be issued mutually.

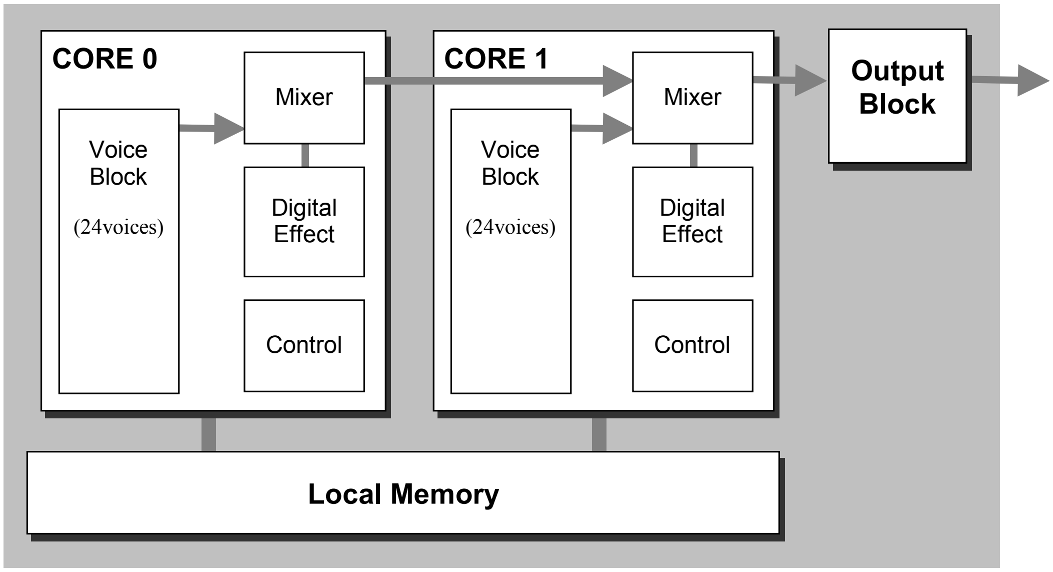
SPU2

The SPU2, which consists of two cores, is a 48 kHz sound synthesizer processor with local memory and external input/output.

As with the PlayStation SPU, each of the two cores (CORE0, CORE1) provides 24-voice voice processing, reverb processing, and master volume processing to generate stereo 2-system (Dry/Wet) audio. 16-bit sound data can also be transferred from the host processor (IOP) and output directly or mixed with voice output. Conversely, the sound data generated by the cores can be transferred to the host processor as well.

The output from CORE0 serves as input to CORE1 and the sound mixed by CORE1 passes through an output block and is output as analog/digital two-system audio.

Figure 7: SPU2 Block Diagram



Data transfers with the host processor (IOP) are performed through DMA or I/O transfer. Registers and interrupt functions are provided to allow the host to query the status of processing, which allows appropriate timing to be used for data transfers. Also, in order to reduce streaming overhead, an Auto DMA transfer function that supports hardware double buffering is provided.

As with sound generation processing, these interrupts and DMA transfers can be performed independently by each of the two cores.

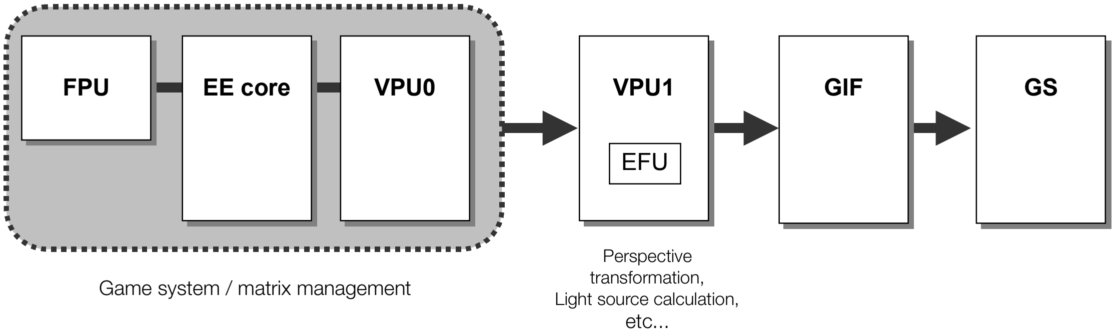
Dataflow

Dataflow For 3D Graphics

3D graphics data generation can be distributed between the EE core+VPU0 and VPU1, but, as was described above, work can be performed either serially or in parallel. When serial distribution is used, the EE core+VPU0 would first perform game system management and geometry calculations, then VPU1 would perform polygon data transformation.

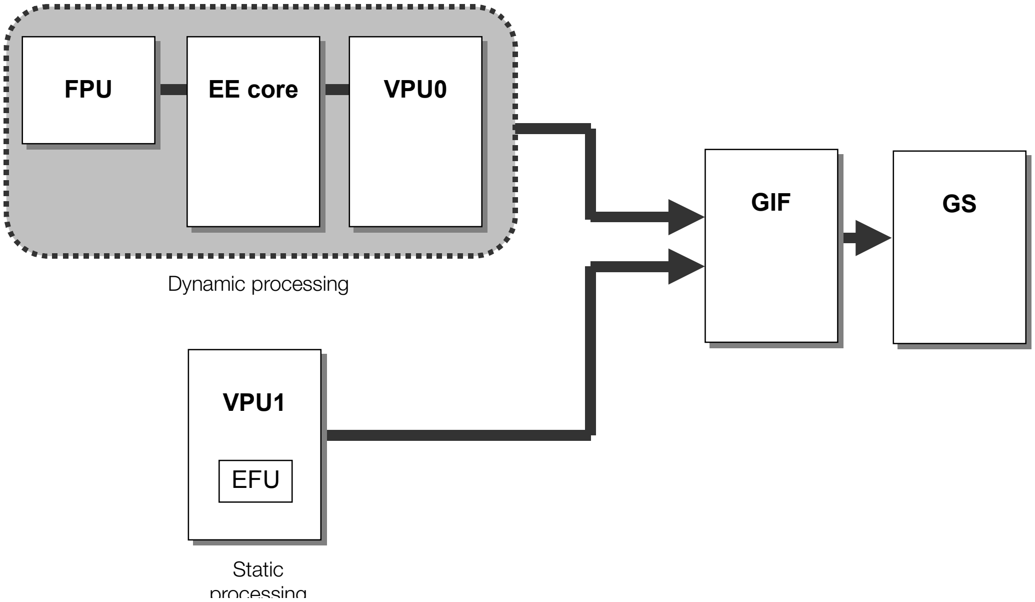
In order to support serial dataflow, a feature called MFIFO is provided in the DMAC. MFIFO works as a ring buffer in one portion of main memory, flowing data from the EE core to VPU1. This method of dataflow is also used in the "Graphics Framework" that is provided as a comprehensive sample or framework of graphics processing.

Figure 8: Serial Dataflow


FPU
EE core
VPU0
VPU1
EFU
GIF
GS
Game system / matrix management
Perspective transformation,
Light source calculation,
etc...

When parallel distribution is used, the EE core+VPU0 would be performing complex, dynamic operations such as character animation, while VPU1 would be simultaneously performing static operations such as background processing.

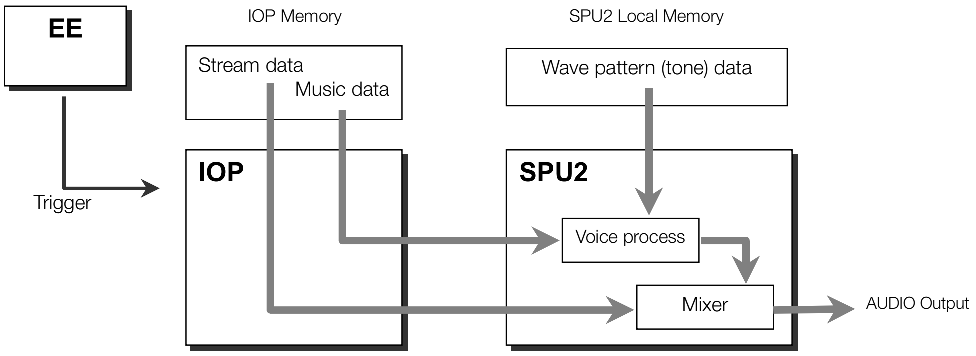
Figure 9: Parallel Dataflow



Sound Data Dataflow

Many different formats can be considered for sound dataflow. Figure 10 shows the processing of a large section using the IOP and SPU2. A format that provides a trigger from the EE for synchronizing the progression of a game is recommended.

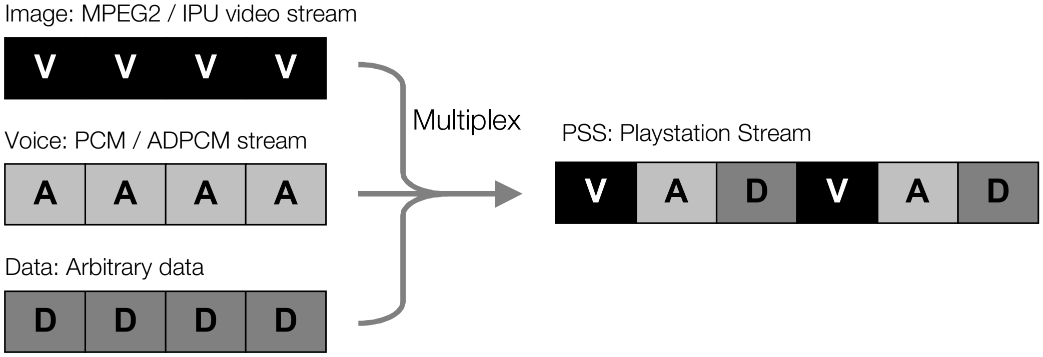
Figure 11: Sound Data Dataflow



Stream Data Dataflow

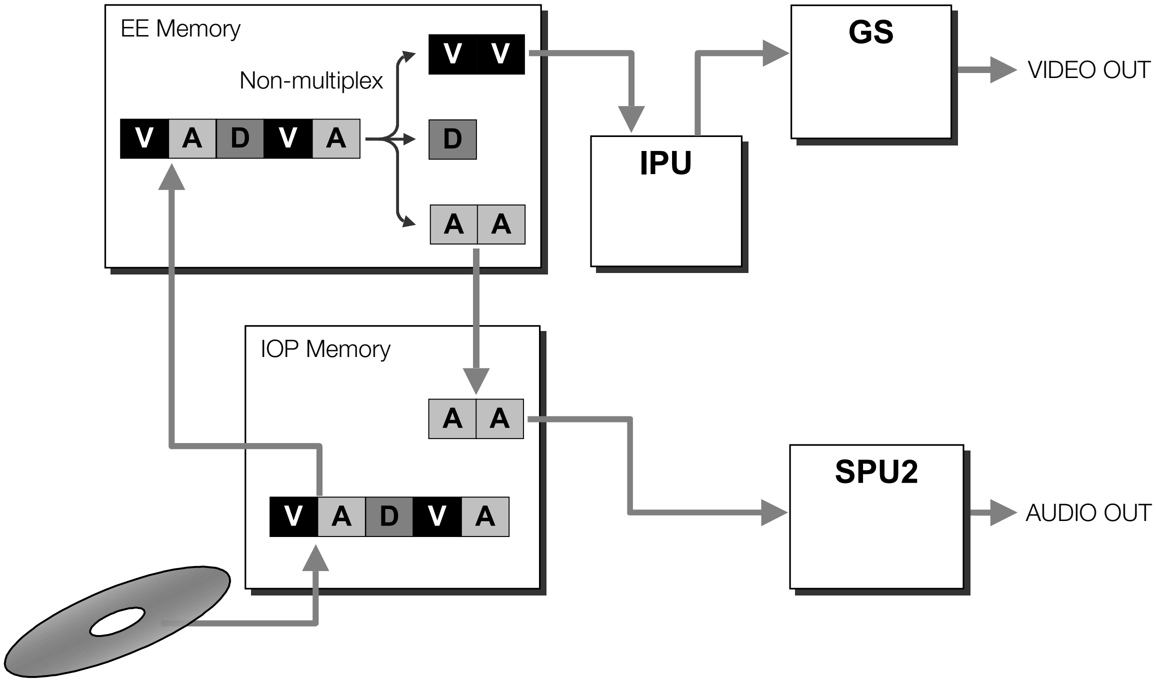
PSS is provided as a stream data format in the PlayStation 2. The PSS format takes arbitrary types of data, such as animation and sound, and multiplexes them together based on the MPEG2 framework. A stream converter tool (ps2str) is also provided for creating data in the PSS format. The stream converter runs under Linux and Windows.

Figure 12: Multiplexed Stream



Other functions are also provided that extract element streams from multiplexed PSS data. These functions are provided as an EE library. Extracted animation streams are decoded by the IPU and EE core and output from GS. Sound streams are temporarily returned to the IOP and then output from SPU2.

Figure 13: PSS Stream Dataflow


EE Memory
Non-multiplex
V
A
D
V
A
V
V
D
A
A
IPU
GS
VIDEO OUT
IOP Memory
A
A
V
A
D
V
A
SPU2
AUDIO OUT