



SQUID Technology Overview

SQUID TECHNOLOGY OVERVIEW

Multimedia applications have evolved from advanced features in consumer devices/mobile phones to a stage where they have become default features that encompass a broad range of products. With the advent of mobile internet, mobile TV and the availability of huge bandwidth through broadband networks, market demand for new devices such as MID's, PMP's and IPTV set top boxes are growing rapidly. Similarly, social networking, VOD and user generated video content have created a need for generic video processing architectures that support higher video resolutions such as SD and HD and flexibility to host multiple applications like video playback, recording, transcoding and editing. Most of the battery operated devices use fixed function video ASICs that tend to restrict application utilization and feasibilities by virtue of a obtaining a power consumption advantage. Conversely, programmable video architecture designs which consider area optimization, power optimization and programmability, and use proprietary data flow processing techniques, brings the advantages of flexibility to incorporate new features and less power consumption equivalent to ASIC implementations. A programmable video architecture provides the flexibility and capability to extend new applications such as editing and transcoding apart from video playback and recording.

Design Goals

To bring the capabilities such as high resolution video processing and support for various features, SoC designers must optimize the power consumption of all components with in the application. Keeping in view of the present and future application demands low power and flexibility are the key requirements for the video processing IP.

Low Power Performance

Video applications are considered as the most power-hungry applications because of its complexity and higher memory bandwidth requirement. To achieve the best power performance, low power process technologies having long gate delays will be used for chip manufacturing. Higher clock frequency can't be met by using these low power process technologies. This pushes the video application should run with lesser clock frequencies and also with less complex functionality in each pipeline stage of implementation so that chip can be manufactured with low power process technologies and operate with less voltage which makes the power consumption to be low.

Video applications at lower clock will be achieved only by using the multiple cores/processors and by running these cores/processors in pipelined manner concurrently. Hardwired solutions achieve the lower clock by using the same principle of having concurrently running hardware blocks for various modules in video application.

Flexibility or Programmability

As explained devices are becoming multifunctional and multiple applications are targeted around video. With the chip manufacturing costs are becoming higher and higher at high end process technologies, consumers are looking for continuous improvements, as multimedia is always coming with new formats, and innovative applications are developed over the period by application developers, solutions will require flexibility to program for new applications.

Programmable multi-core solutions can bring the advantages of both flexibility and low power, and making the solution future proof. This can be achieved by designing the architecture with optimally designed programmable modules targeting for specific functionality of video applications. By programming these modules to perform specific tasks and connecting those modules in pipeline will achieve required applications with lower clock. In other words, the processing architecture has to execute different parts of an algorithm (e.g. Stream parsing, Transforms, Motion-compensation, De-block filtering in an H.264 video decoder), and different operations within each module (additions, multiplications, memory accesses etc) concurrently in a pipelined manner.

Multi-core programming Complexity

Programming complexity and application development will become exponentially complex as the number of cores increases. To select any multi-core solution this is a critical requirement to understand the simplicity of application development. To make the programmability easier, number of functional units should be as less as possible and if each functional unit is capable of handling a relatively big task of the video application in pipeline, application development will be easier and simpler. Architectures can be designed with several tiny cores, but the partition of the application to several tiny cores, synchronizing and scheduling the overall application becomes very cumbersome and difficult. Because of the inherent dependencies within the algorithm, application can't be efficiently partitioned to so many cores and which sets a limit to which an application can be parallelized.

Using an extremely parallel architecture with simple operations makes it possible to use a low clock frequency, which in turn makes it possible to use a slow, low-leakage silicon process, and to lower the supply voltage level of the video IP block in a SoC. As power has a square dependency on voltage, this gives a huge advantage over designs that require a higher clock-frequency.

Hardwired Solutions

Hardwired Solutions for video applications comes up with having multiple small blocks of implementations such as stream parsing, motion compensation, prediction, transform and de-blocking filter kind of modules. As the number of formats of video to be supported is growing, the number of blocks will keep on growing in the hardwired design and slowly the size of the video processing IP if it wants to support couple of video encoders and decoders. These designs will be able to manage the power by clock gating and power gating methodologies but the overall size and area will become big and at the end SoC developers do not have a choice to modify the functionality or improve the algorithms etc. Support for other post processing and camera processing is not possible on the same processing sub-system. Slowly it becomes a cumbersome design to support the real world requirements.

Multi core Programmable video architecture (SQUID):

Squid presents an optimally designed scalable, programmable video architecture with multiple functional units which enables to realize the video applications with low clock and meet the requirements of both battery operated and consumer electronics devices.

Video applications involve the following operations considering the major chunk of operations.

- Control sensitive data flow processing including Single operation on multiple data like SIMD operations such as transforms, filters etc
- Sequential control plane tasks such as parsing, motion vector prediction etc
- Sum of absolute differences computation for motion estimation
- Data movement across various functional units

Each type of processing category comes up with different type of computational needs. A single processor core to handle all types of tasks will make the architecture inefficient and increases the clock requirement. Each of the processing type constitutes considerable amount of the complexity in video applications. Having a specifically targeted programmable core for each type of processing block will enable efficient handling of each computational block and having efficient data interconnect across these processing cores will enable an efficient programmable solution. Above four types of tasks constitutes any typical video application including, video encoding, decoding, pre-processing or post processing. Squid's solution comes with a different processing core to handle each type of computational requirement and provided an efficient interconnect to avoid unnecessary data movements. This architecture meets the challenges of the video SoC developers meeting all the primary requirements of IP power consumption, area/foot-print and flexibility. A block consisting of all these cores is called "**vivid**" and becomes the building block of the Squid's video processing intellectual property.

Vivid architectural building block of the Squid's IP consists of the following cores

- **Data flow processor**
- **Bit-Stream Processor**
- **Control processor**
- **SAD (Motion Estimation) accelerator**
- **DMA Controller**

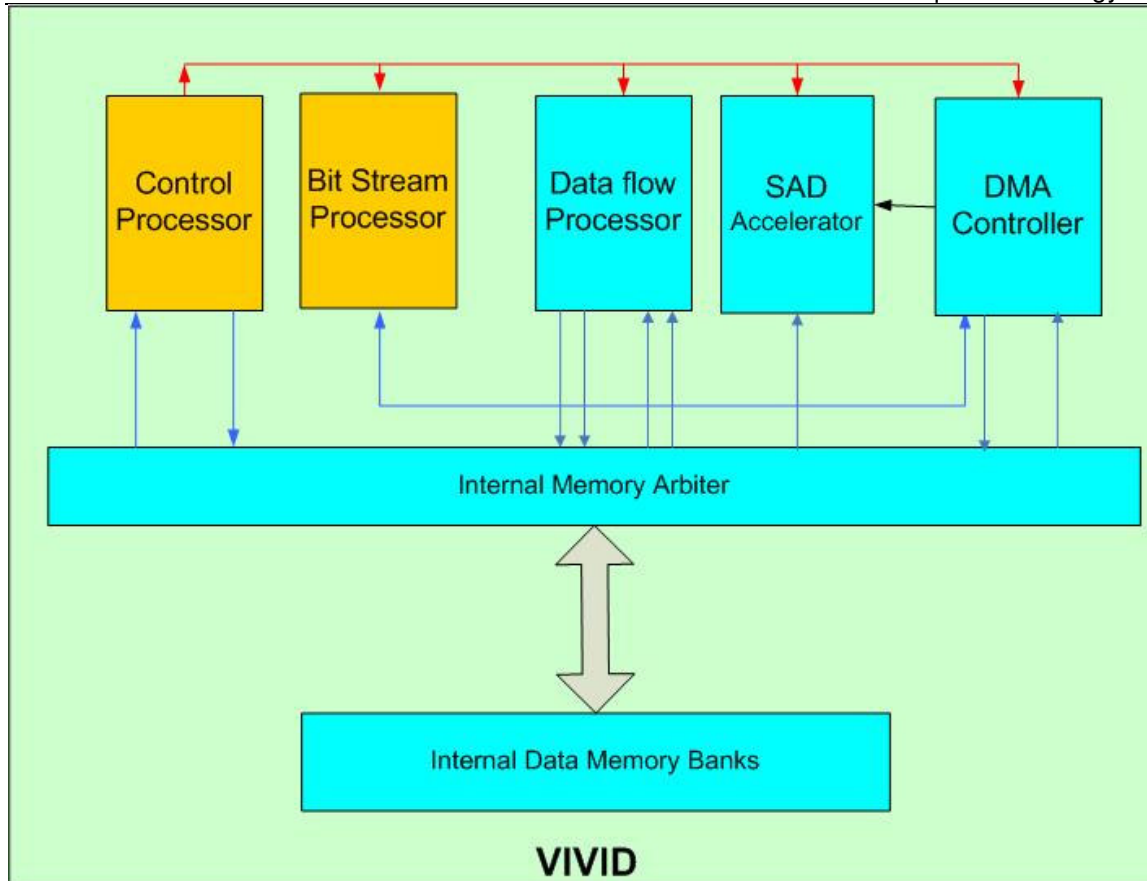


Figure 1 Vivid

The combination of above gives balanced control and data flow performance, DMA for no-overhead communication across various banks of memory in vivid and memory transactions with external memory. SAD accelerator is a critical block for any video processing applications which helps to implement any type of motion estimation algorithm through its unique programmable model.

Data flow processor is based on Squid’s proprietary technology. It runs without a program controller, runs based on static scheduled data flow processing, and provides the data flow performance required for video applications. Data flow processor consists 8 16-bit ALU’s and an efficient interconnect to connect these ALU’s. 16-bit ALU’s can also work in SIMD mode with two 8-bit data operands. Data flow processor performs efficiently any control sensitive data flow tasks such as transforms, motion compensation, de-blocking filters and intra prediction computations in H.264 decoder. Dataflow processor’s performance metrics matches the ASIC performance metrics (such as clock requirement and power consumption) for any algorithm. Squid’s proprietary technology ensures that all the ALU’s in the data flow processor are efficiently utilized and provides 100% occupancy in almost all the algorithms. 70% of the overall video application will be handled by the data flow processor.

Bit-stream processor is a 3 way VLIW processor meant for stream processing and very efficient control processing. Proprietary architecture of bit-stream processor will achieve 4x performance compared to ARMv6 architecture for any control intensive tasks. Bit-stream engine in the bit-

stream processor handles any complex variable length coding and control intensive operations efficiently.

Control processor is the master processor of the vivid which schedule the tasks across various other functional units and synchronize the activities. Control processor also interfaces with the host processor. Other processors within the vivid are managed by control processor which triggers the tasks and manages the resources using interrupts from other cores within the vivid. Control processor is an interrupt efficient 16 bit micro controller.

SAD accelerator is a Sum of Absolute Differences accelerator which is one of the heavily performed computations in motion estimation process of any video encoding/de-interlacing application. Squid's SAD accelerator uses the same data memories available within the vivid and it has several configuration options to use as per the application demands. User can program it to implement any type of motion estimation algorithm such as full search, hexagonal search and any other sub-optimal fast decision making methods.

A vivid block consisting of multiple functional units requires an efficient memory subsystem to reduce the unnecessary data movements across different units and simultaneously functional units should not face the memory bottlenecks, which will harm the performance. Vivid's memory subsystem consist 6 banks of memory to reduce the memory access bottlenecks for each functional unit. All the memory blocks are accessible by all the functional units with a fixed priority order in the case of contention.

DMA controller is a multi channel direct memory access controller which can transfer the memory contents between internal data memory banks of vivid and with the shared memory for external memory transfers.

Vivid Benchmarks

A vivid is able to clock more than 300 MHz on TSMC 90nm low power process and its area is smaller than 4 mm² including memories and logic. H.264 decoder at SD resolution requires 100 MHz on a single vivid with power consumption of 40 mw in TSMC 90nm process. Vivid can provide 4.2 GOPS of 16 bit performance or 6.6 GOPS of 8 bit performance clocking at 300 MHz apart from the SAD computations and memory transactions through DMA. Some of the typical benchmarks for video specific computations are shown in Table 1. Benchmarks for the performance of the data flow processor are provided based on the number of basic arithmetic operations including memory load and store operations performed per cycle for video algorithms.

Table 1

Algorithm	Number of Basic operations	Number of Cycles on Data flow processor	Number of operations per cycle
1024 point complex FFT	81920	5120	16
Integer Transform (4 x4)	248	16	15.5
Hadamard Transform	182	14	13
Integer Transform (8x8)	1254	128	9.8
DCT (8 x 8)	1792	128	14
De-blocking filter (one edge)	96	4	24

Quarter Pixel Interpolation (8 x 8)	3694	272	13.5
Random SAD	768	16	48

HD video processing

High definition video processing is achieved by multiple ways using Squid's technology. It can be achieved by simply clocking the vivid at higher clock frequency using better process technologies. Similarly HD solutions also can be achieved by developing a homogeneous multi core architecture based on the vivid as a building block at lower clock frequencies. Figure 2 shows Squid's IP having multiple vivid blocks. Multiple vivid's communicate using the shared memory. Shared memory will be accessed by vivid through the DMA controller only, so that memory hierarchy is clean and simple. As shown in Figure 2, number of vivid blocks and size of the shared memory banks can be chosen based on the application requirement. Squid's IP can be extended to 1080p or beyond performance without any problem by extending multiple vivid blocks or increasing the clock. Squid's multi core IP's will demonstrate a globally homogeneous and locally heterogeneous architecture which are extremely efficient and rated as the best in class performance.

High definition (720p) video processing requires at least 2.5 times more computational power compared to the SD resolution solutions. Squid provides multiple solutions to address 720p resolution video processing based on the customer requirements as SQ100 with single vivid can handle seamlessly 720p H.264 video processing at 225 MHz or Squid's SQ110 architecture with two vivid cores is able to perform 720p H.264 video processing at less than 120 MHz. SoC developers can choose from various IP cores from Squid depending on the choice of size, clock, power and flexibility. Similarly SQ110 can meet the requirements of 1080p H.264 video processing at the clock of 300 MHz or SQ120 is available with 3 vivid cores can handle at 150 MHz. Similarly post processing along with decoders or the pre-processing along with encoders can be easily accommodated with these scalable solutions by having an additional vivid or increasing the clock frequency. Squid's IP supports the upcoming video algorithms such as SVC and HD-JPEG. Same IP can be extended to the 10 bit video processing with minor changes.

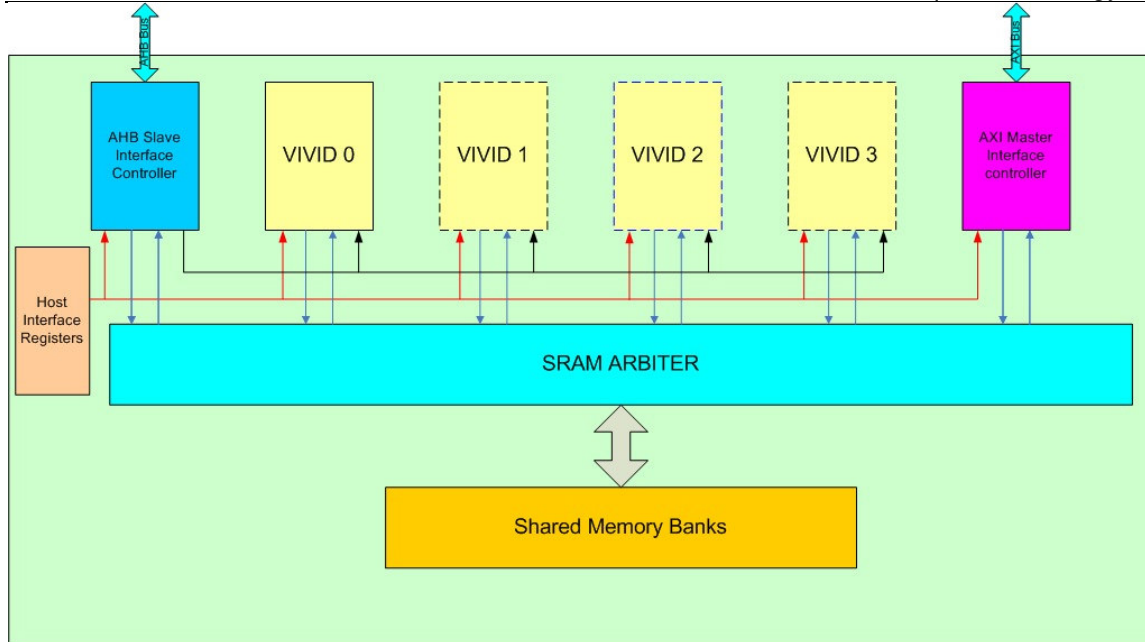


Figure 2 Squid's IP with multiple VIVID blocks

Power Consumption

Architecture is designed in the view of achieving the best power performance by appropriately choosing tailor made processors for each processing block which brings the efficiency. Apart from the basic architecture design, to control the power consumption, all the traditional methods have been employed at RTL level, clock gating, power gating, and applications also are developed to reduce power by avoiding any redundant computations, memory transfers, etc. All the logic can be used with lesser voltage levels to reduce the power consumption.

Conclusion

Squid's IP shows a way to build a low power multi core programmable architecture with an easy application development model by having a balanced control and data flow performance specifically designed for target application. Similar methodology can be employed for other application areas such as graphics and Software defined radio.