Proceedings of the International Conference on Wireless Networks & Embedded Systems (WECON 2009)

Vol. 2of 2

ISBN-13:9788185266657
ISBN-10:81-85266-65-4

[With contribution and editing by Prof. T.L.Singal and Prof. Isha Verma]

En

Published by: *Mohindra Capital Publishers, 25/7, Industrial Area, Phase-II, Chandigarh-160002. Ph.No. 0172-5077345 on behalf of Chitkara Institute of Engineering and Technology, Chandigarh-Patiala National Highway, Tehsil-Rajpura, Distt. Patiala-140 401(INDIA)*

Computer Vision in an Embedded System

Vishal S.Vora- Pg Student, PG Student-Yagnesh N. Makawana, PG Student-A. M. Kothari, Prof.- Mr.A. C. Suthar
Deptt of E&C, CCET, Wadhwan, Gujrat, India. vsvora@aits.edu.in

Abstract- **In this paper describes how the embedded systems community with an insight into computer vision challenges and techniques, basis on algorithmic and hardware specific considerations when "outsourcing" computation to a DSP, FPGA, or onto an embedded platform, and provides guidelines for how to best improve the, runtime performance of computer vision applications. Because of Embedded computers and computer vision systems are two of the largest growths markets in the industry "Systems on a chip" is more popular. because of the power, cost and space-saving combination of previously external chips onto the same die, including analog-to-digital converters, bus interfaces, I/O controllers, and even analog components. By using these integrated processing capabilities, computer vision has become a viable and preferable solution to many legacy and novel problem settings in industry, military, and consumer markets.**

Keywords—Digital Signal Processing, FPGA, System On Chip, Embedded Systems.

I. Introduction

Basically an embedded system is a set of digital and possibly analog circuitry that has a precisely defined purpose. It has one or multiple processors that serve a dedicated purpose for one particular system, optimized for a particular function to achieve shorter startup times, higher processing speed, greater reliability, lower cost, lower power consumption and/or some other property that a general-purpose computing system does not fulfill. The software is often called firmware, emphasizing its inseparability from the hardware. In an embedded system core part is microprocessor, usually surrounded by memory chips, input/output controllers, co-processors, analog components, and any imaginable electronic component.

Embedded system are particularly well suited to process streams of data at high speeds with relatively small software programs, such as video processing in television sets and DVD players, real-time control tasks in cars and airplanes, and cell phone signal processing, but also in microwave ovens, digital parking meters and pace makers. For vision, A *smart camera* is a combination of an imaging device and a computational unit whose combined output is a processed video stream, such as the location and identity of a person. However, integration and miniaturization make it increasingly feasible to develop smart cameras with much smaller form factors, down to single-chip devices that integrate processing power right on the imaging chip.

II. Hardware

For general-purpose to system-specific chips various hardware are useful, some of hardware describe here with its main characteristics.

A. Digital Signal Processor

A digital signal processor is similar to a general-purpose processor in many aspects. It has fixed logic, that is, the connections between logic gates can not be changed. It provides a fixed instruction set (ISA) to the programmer, and it expects a program in this ISA which it will then execute in a sequential manner. Most DSP ISAs exhibit a similar structure as GPP ISAs, complete with arithmetic and logic instructions, memory access, registers, control flow. DSP's instruction set is optimized for matrix operations, particularly multiplication and accumulation traditionally in fixed-point arithmetic, but increasing also for double-precision floating point arithmetic. DSPs exhibit deep pipelining and thus expect a very linear program flow without frequent conditional jumps. They provide for SIMD instructions, assuming a large amount of data that has to be processed by the same, relatively simple, mathematical program. Many DSPs are VLIW architectures, the types of instructions that are allowed together within one VLIW will be executed in parallel depends on the function units that can operate in parallel.

DSP programs are relatively small programs with few branch and control instructions, as opposed to entire operating systems running on general purpose CPUs. Since DSPs usually execute small programs on huge amounts or endless streams of data, these two pieces of information are stored in separate memory blocks, often accessible through separate buses, many DSPs provide on-chip ROM for program storage, and a small but efficient RAM hierarchy for data storage. An embedded system also includes a separate non-volatile memory chip such as an EEPROM or flash memory.

DSPs lack just a few operations, mostly operating-specific instructions, but they can perform them faster, they need less power, they dissipate less heat, they have short start-up times, they can operate in a larger temperature range, and they are less expensive because the chip contains only the necessary components.

B. Field Programmable Gate Array

A Field Programmable Gate Array, or FPGA, is a semiconductor in which the actual logic can be modified to the application builder's needs. The chip is a relatively inexpensive, off-the-shelf device that can be programmed in the "field" and not the semiconductor fabrication. It is important to note the difference in software programming and logic programming, or logic design as it is usually called: a software program always needs to run on some microcontroller with an appropriate instruction set architecture (ISA), whereas a logic program *is* the microcontroller.

Some of the modern FPGAs integrate platform-or hard multi-purpose processors on the logic such as a PowerPC, ARM, or DSP architecture. Other common hard and soft modules include multipliers, interface logic, and memory blocks. Some more complex FPGAs can modify and reprogram their general-purpose logic blocks even on the fly, that is, while another part of the chip keeps running.

The logic design determines the FPGA's functionality; there are three types of FGPAs: antifuse, SRAM, and FLASH. Antifuse chips are not re-programmable. FLASH (EPROM) is also non-volatile, meaning that the logic design stays on the chip through power cycles. It can be erased and re-programmed many times. SRAM programming on the other hand is volatile – it has to be programmed at power on.

C. *Application Specific Integrated Circuits*

An Application-Specific Integrated Circuit (ASIC) is a chip that is designed and optimized for one particular application. The logic is customized to include only those components that are necessary to perform its task. Even though modules are reused from ASIC to ASIC just like FPGA modules, a large amount of design and implementation work goes into every ASIC. Their long production cycle, their immensely high one-time cost, and their limited benefits. Contributing to the high cost, they need to be respun if the design changes just slightly, costing months and usually hundreds of thousands of dollars.

D. *System on Chip*

A System on Chip (SoC) contains all essential components of an embedded system on a single chip. Sometimes only refers to the digital components and analog components. Most SoCs have a GPP such as an ARM, MIPS, PowerPC, or an x86-Based core at their heart, supplemented by a DSP. Standard microcontroller components (busses, clocks, memory), typical integrated components like, Ethernet MAC, PCMCIA USB 1.0 and 2.0 controller, Bluetooth, RS232 UART, IrDA, IEEE 1394 (FireWire) controller, display interface, flash memory interfaces ADCs and DACs Systems on a chip are of particular interest to highly-integrated devices like mobile phones, portable DVD and mp3 players, set-top-boxes and cable modems. Many SoCs have dedicated circuitry for video processing, which usually means hardware support for decoding from (and sometimes for encoding into) the various video formats, including MPEG2, MPEG4, and H.263.

E. *Smart Camera Chip and Boards*

Processors, SoCs, and chip sets that directly support higher-level computer vision tasks come in various flavors, from CMOS image capture chips that include one or multiple small processors to frame grabber PCI boards with multiple full-scale CPUs.

III. Software And Programming Tools

Software tools that is necessary or helpful to implement those programs and to transfer them to the embedded system.

A. *Language and Operating Systems for DSP*

DSPs are usually programmed in C at first, followed by machine code optimization for critical parts. A DSP rarely just executes one tiny program on an endless stream of rather uniform data, but instead has to perform some general tasks occasionally. Thus, it is usually controlled by a DSP operating system. A large number of companies offer an even larger number of them, frequently classified as a real-time OS. Linux is a common choice due to its flexibility, particularly on Systems-on-Chips. Following are some of your choices for operating for DSPs and/or SoCs. One of the main characteristics of these OSs is their small footprint, typically only one to tens of MB.

B. *FPGA Logic Design*

Determining the function of each of the logic elements on a Field-Programmable Gate Array as well as their interplay and connectivity is called the logic design or digital design.

1) Design Flow: The primary block of an FPGA is called a logic element, which usually functions as a lookup-table and contains a fancy flip-flop and a few logic gates. The FPGA needs to know how to arrange these. The digital design needs to be mapped to logic elements, this is called technology mapping. The netlist is the actual file that gets mapped and routed to a specific FPGA.

The (logic) synthesis translates design into a so-called netlist. This is usually a heavily optimizing synthesis that not only optimizes the HDL with traditional compiler means, but also makes use of hard macros, chip specifics etc. The netlist is usually in the Electronic Design

Interchange Format, or EDIF. This entire process is also called FPGA technology mapping.

Next, the netlist needs to be written to the FPGA. This process is called place-and-route. First, the netlist's component references (registers, logic, macros) are assigned to FPGA locations. This is obviously a manufacturer-dependent process as FPGA layouts vary considerably. The result is a FPGA bitmap, which is sent to the FPGA via a number of different means, the most popular being JTAG.

2) Hardware Description Language: A Hardware Description Languages, or HDL [1], describes the behavior of a chip. Many companies offer a product that combines all necessary and a host of extra tools for FPGA programming into one software package. This is referred to as electronic design automation.

One essential component that EDA tools provide are hardware libraries or hardware macros, containing the building blocks for common functionalities like memory, signals, multipliers, and all the way up to entire DSP and GPP soft cores, which are to be created out of generic FPGA logic elements.

3) JTAG and Boundary Scan: The interface to generate tests and access test results on printed circuit boards was standardized by the Joint Test Action Group and has since itself become known as JTAG. The type of testing that JTAG was intended to perform is actually called boundary scan. It works by accessing the input and output lines of a chip or circuit board and storing their values in a boundary scan register. To facilitate this, a JTAG-compliant chip or board has additional logic around a chip's actual core. This logic observes input values to the core and writes them serially to a dedicated test access port, or TAP, where they can be compared to the input signals that were applied to the chip's

physical connectors. The TAP is a 5-pin serial connection through which all communication with the boundary logic is sequenced to or from the debugging facility, usually a PC. JTAG is frequently used as the method of choice to route the actual program (the netlist/bitmap) to an FPGA. While a convenient and wide-spread access method, JTAG is not suited for high-bandwidth debugging.

IV. ALGORITHMS

We present some of the most important real-time algorithms from different fields that embedded vision relies on: data analysis, digital signal and image processing, low-level computer vision, and machine learning.

F. Tensor Product

For a very important class of algorithms there is a theoretical tool that can be of use in comparing different algorithms as they map on different processor architectures [2]. This tool applies to digital filtering and transforms with a highly recursive structure. Important examples are [3]:-

1) Linear convolution, e.g., FIR filtering, correlation, and projections in PCA (Principal Component Analysis)
2) Discrete and Fast Fourier Transform
3) Walsh-Hadamard Transform
4) Discrete Hartley Transform
5) Discrete Cosine Transform
6) Strassen Matrix Multiplication

To investigate these algorithms, the required computation is first represented using matrix notation. This algorithm efficiently implemented on a SIMD architecture. This method does not take into account non-deterministic effects of cache memory and super scalar issue parallelism. Same formulas can be used to determine many mathematically algorithm which are suitable for a particular processor architecture.

G. Sorting Algorithm

A common task in data analysis is sorting of data in numerical order. The fastest sorting algorithm is Sir C. A. R. Hoare's Quicksort algorithm [4]. A slightly slower J. W. J. Williams' Heapsort algorithm [5} but in-place sorting and has better worst-case asymptotics. In Table 1 shows some more information on asymptotic complexity.

H. Golden Search Algorithm

It is a simple but very effective algorithm to find a maximum of a function. Basic assumption is that getting the value of the function is expensive because it involves extensive measurements or calculations [6]. By using some application-specific method It is a one, and only one, maximum in the interval given to the algorithm.

I. Kalman filtering

A Kalman filter [7] is an estimator used to estimate states of dynamic systems from noisy measurements. This method is attractive for the recursive solution and suitable for real-time implementation on a digital computer. The fact that the solution is recursive means that at each time instant updating the previous estimate forms the estimate using the latest measurement. If there was no recursive solution, we would have to calculate the current estimate using all measurements. Other, more complicated formulations of Kalman filter allow colored noise, others can account for unknown system model parameters, and some can deal with nonlinearities. In recent years, a more general framework called particle filtering has been used to extend applicability of Kalman's approach to non-Gaussian stochastic processes.

J. Fast Fourier Transform

Fast Fourier Transform (FFT) is a common name for a number of $O(n\log n)$algorithms used to evaluate the Discrete Fourier Transform (DFT) of n data points. Several ways the FFT can be used to speed up seemingly unrelated operations, such as convolution, correlation, and morphological operations [8]. Some of the other fast algorithms for convolution that are non-FFT based like fast convolution and correlation, windowing and data transfer, fast morphing.

K. PCA Object Detection and Recognition

Principal Component Analysis (PCA), also known as eigenpictures or eigenfaces [9, 10], is a popular computer vision technique for object detection and classification, especially in the context of faces. The most common PCA classification scheme is based on the reconstruction error. If PCA is implemented using a set of orthonormal eigenvectors, this implies use of the floating-point precision. On a TI DM642 processor, this technique has reduced the execution time of PCA classification by a factor of seventeen. The execution time is further reduced when utilizing the SIMD operations available on the chip. These operations utilize 32-bit multipliers and adders to do four simultaneous 8-bit operations. To estimate the numerical error introduced by fixed-point representation the quantization error as uniform noise with variance $12222XmBe-=\sigma$ (1) where B+1 is the number of bits in the representation. As a result of going from a 32-bit floating-point representation to an 8-bit fixed-point representation, the SNR change due to quantization error is Δ(SNR) = 20log2$B0-B1\approx$ -144dB, where $B0$ = 8bit and $B1$ = 32bit. By itself this may seem like a large degradation, but we are still left with around 48 dB of SNR. Object detection is a related but different problem of finding candidate images to be used in recognition. PCA can be used for detection, when the entire scene is scanned by sliding the PCA classifier in search of objects such as faces. In this case, projections turn into correlations. As we discussed earlier in this section, fast correlation algorithms exist, similar to fast convolution algorithms, based on FFT or other approaches.

L. Suitability of Vision Algorithm

Some image processing and computer vision methods are better suited to execution on a DSP or FPGA than others, for different reasons. The theoretical, potential speedup of common algorithms to be placed in a DSP or FPGA, and try to generalize and look for the reasons. In general, these are

properties that make an algorithm a good candidate to be executed in special hardware:
1) Streams/sequential data access, as opposed to random access
2) Multiple, largely independent streams of data
3) High data rates
4) Data packet size is fixed, or at least bounded by a small number
5) Stream computations can be broken up into pipeline stags, i.e. the same (possible parameterized) computations independent of prior stage's output
6) Fixed-precision values (integer or fix-point fractions)
7) Analyzable at instruction and module level, that is, little between-stream communication necessary

Consider what really constitutes digital video data. Video data is dense 2D data with each element of a constant size. Most frequently, it is streamed along a communication bus in a line-by-line manner, or interlaced in the case of NTSC or PAL video sources. Traditionally, data is buffered and processed whenever the program is the proper instruction. But ideally suited to processing streamed data are "online algorithms" which process data as they arrive. This online or data-driven execution is advantageous for low latency and low buffering requirements. However, it might result in unnecessary computations as it is not usually known a-priori what data are needed in the end. A demand-driven approach avoids this, but incurs higher latency due to its reverse dependency chain.

Computer vision methods commonly entail linear algebra, particularly for non-image data such as feature vectors for classification. Thus, matrix operations are common, including matrix multiplication, calculating the inverse and transpose, Gaussian elimination, residual minimization for linear systems, singular value decomposition, and eigenvector and eigen value Calculations.

V. CONCLUSION

Computer vision on embedded platforms and building smart cameras is an exciting field with tremendous growth prospects. In this paper on hardware and software that are most often found in embedded systems, with a focus on architectures and tools that are built for computer vision applications. With a particularly difficult issue real-time algorithms for common computer vision operations. These are great times to witness exceedingly rapid progress in a field that had in the past had difficulties living up to overly great expectations. Yet now computer vision has made inroads in many applications, from games to surveillance, from smart airbag deployment to automatic parking controllers for automobiles. Computer vision has the opportunity to sense beyond the confines of a computer, and embedded systems have the power to make the computer itself mobile. In combination, these technologies harbor possibilities that can only be imagined as the first steps are taken into this exciting field!

VI. REFERENCES

[1]. J. Bhasker. *A VHDL Primer*. Prentice Hall, 3 edition, 1999.

[2]. J. Granata et al. The tensor product: A mathematical programming language for FFTs and other fast DSP operations. *IEEE Signal Proc Mag*, pp 40–48, 1992.

[3]. J. Granata et al. Recursive fast algorithm and the role of the tensor product. *IEEE Trans Signal Proc*, pp 2921–2930, 1992.

[4]. C A R Hoare. Quicksort. *Computer Journal*, pp 10–15, 1962.

[5]. J W J Williams. Algorithm 232 (heapsort). *Comm ACM*, pp 347–348, 1964.

[6]. EKPChongandSHZak. *An Introduction to Optimization*. Wiley, 2001.

[7]. R E Kalman. A new approach to linear filtering and prediction problems. *ASME J of Basic Engineering*, pp 34–45, 1960.

[8]. J W Cooley. How the FFT gained acceptance. *IEEE Signal Proc Mag*, pp 10–13, 1992.

[9]. L Sirovich and M Kirby. Low-dimensional procedure for the characterization of human faces. *J Opt Soc Am A*, pp 586–591, 1987.

[10]. M Turk and A Pentland, Face recognition using eigenfaces. *Proc CVPR*, 1991.