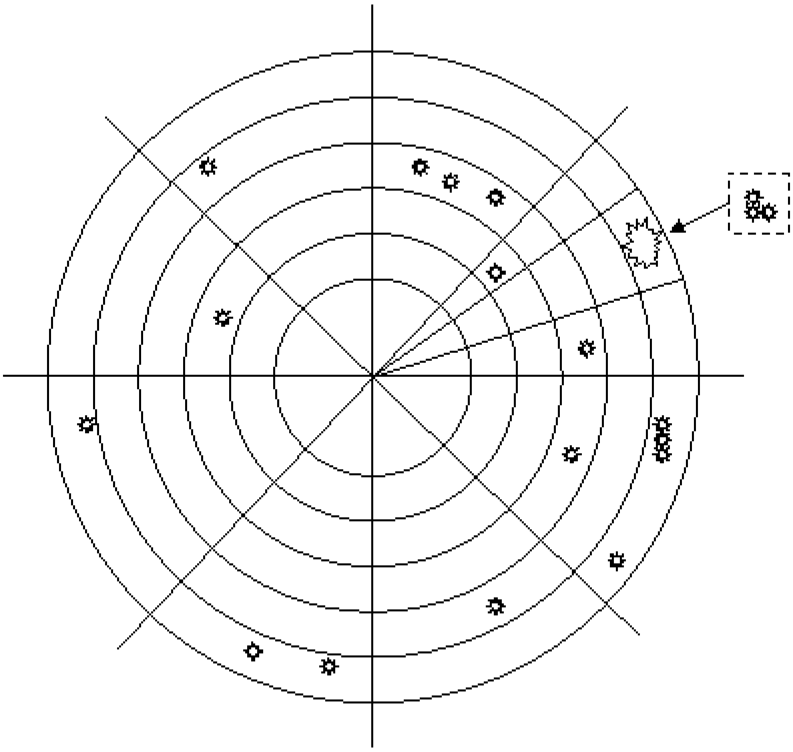
CHALMERS



Processing solutions for high-performance in combination with specific radar algorithms

Master of Science Thesis in the Programme Secure and Dependable Computer Systems

CHRISTIAN K.M. STRID

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
Göteborg, Sweden, March 2011

Processing solutions for high-performance in combination with specific radar algorithms
For Saab AB at Saab AB, Electronic Defence Systems, Gothenburg.

CHRISTIAN K.M. STRID

Examiner: Lars Bengtsson
Supervisor at Saab: Lars Henricson, Patrik Kanthe, Per Ericsson

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden March 2011

Abstract

Radars today set high demands on underlying computers that perform signal processing. There are different ways of handling these requirements with varying combinations of hardware and software, some specialised for one or a few specific tasks while others are capable of more general types of tasks.

The processing board which handles high-resolution radar today within some of Saab’s radar products has become a subject for upgrade and new solutions are being investigated to replace present platform.

This thesis aims to evaluate new processing solutions for future upgrades, solutions capable of handling increasing workload and new functionality.

Results indicate that the latest of Intel’s processor architectures are more than capable of handling the required workload for relevant radar systems, without any changes to the high-resolution algorithms and only small changes to software architecture.

Sammanfattning

Dagens radarprodukter ställer höga krav på underliggande datorer som utför signalbehandling. Det finns olika sätt att hantera dessa krav med varierande kombinationer av hårdvara och mjukvara, vissa specialiserade för en eller några specifika uppgifter medan andra är kapabla att hantera mer generella uppgifter.

Processor korten som idag exekverar hög-upplösnings radar inom några av Saab's radar produkter ska uppgraderas och nya lösningar undersöks för att ersätta nuvarande plattform.

Detta examensarbete syftar till att undersöka och evaluera nya processor lösningar för framtida uppgraderingar, lösningar som klarar av ökande arbetslast och ny funktionalitet.

Resultaten från detta projekt indikerar att Intel's senaste processor arkitekturen är mer en kapabla till att hantera den krävda arbetslasten för relevanta radar system, detta utan några förändringar av algoritmer och endast små förändringar av mjukvaru-arkitekturen.

Contents

Abbreviations

Altivec	A vector instructions extension
AMP	Asymmetric Multi-Processing
API	Abstract Programmer Interface
AVX	Advanced Vector eXtension
CPCI	Compact Peripheral Component Interconnect
CPU	Central Processing Unit
DP	Data Processing
DSP	Digital Signal Processor/Processing
FC	Fire Control
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
GPU	Graphic Processing Unit
GP-CPU	General Purpose Central Processing Unit
GP-GPU	General Problem Graphics Processing Unit
GPIO	General Purpose Input Output
GFLOPS	Giga Floating point Operations Per Second
Gbe	Gigabit Ethernet
GB	Giga Byte
HR	High Resolution
HRFC	High Resolution Fire Control
HD	High Definition
HRE	High Resolution Extractor
IFFT	Inverse Fast Fourier Transform
I/O	Input/Output
MUSIC	Multiple Signal Classification
MB	Mega Byte
MIMD	Multiple Instruction, Multiple Data
MMI	Man-Machine Interface
OS	Operating System
PRF	Pulse Repetition Frequency
PCI	Peripheral Component Interconnect
PC	Personal computer
PMC	PCI Mezzanine Card
PCI-X	Peripheral Component Interconnect eXtended
PCIe	Peripheral Component Interconnect Express
PPC	Power PC
RC	Radar Controll
RTOS	Real-Time Operating System
RTAI	Real-Time Application Interface
RBS	Range Bin Selector
RX	Receiver
SIMD	Single Instruction, Multiple Data
SBC	Single Board Computer
SP	Signal Processing
SF	Sample frequency
SPE	Signal Processing Element
SDK	Software Development Kit
SMP	Symmetric Multi-Processing

SVD	Singular Value Decomposition
SSE	Streaming SIMD Extension
SISD	Single Instruction, Multiple Data
TX	Transmitter
VME	Bus standard, ANSI /IEEE 1014-1987
VPX	VME based bus standard, ANSI/VITA 46.0-2007
VSIPL	Vector Signal Imaging Programming Language
XMC	Switched Mezzanine Card

1. Introduction

Processing solutions for high-performance in combination with specific radar algorithms.

1.1. Background

Modern signal processing requires high performance and high throughput. Large parts of data can be filtered out before reaching the actual detection intelligence within the signal processing. But still there are large amounts of data left to be processed by radar algorithms to detect targets.

These algorithms perform high-intensity complex calculations which allow the signal processor to detect targets and discard noise, clutter and Jammers. Saab uses a specific high-resolution algorithm which increases resolution and separates targets which might appear as one with the common detection algorithm. Calculations are based on data which is collected through analysis of the received echoes. Rapid development within computer technology has made it possible to process multiple times the amount of data that was possible ten years ago. More data will allow the algorithm to increase resolution even more and with higher accuracy and with added functionality.

Currently the high-resolutions algorithm executes on a DSP board which uses four PowerPC MPC7410 signal processing elements and one MPC8240 as dispatcher. This board has been used for about a decade and it has been announced to reach end of life within a short future.

The MPC7410 is capable of issuing instructions from specialised instruction sets which perform vector arithmetic's. This increases possible computational throughput multiple times. These instructions are part of extended instructions set named Altivec and the technology is commonly referred to as SIMD.

A decade ago Altivec was the most mature SIMD instruction set available on the market. Today Freescale which produces the 7410 has no new processors under development which are Altivec enabled. In addition there are comparable alternatives to Altivec from Intel which in recent years have shown more and more interest for the embedded market.

With end of life of current Digital Signal Processor-board (DSP-board) closing in and the emerging alternatives to Altivec the opportunity for new technology has opened.

Not only has the high-resolution signal processing board reached end of life but so has also the generic SBC used for SP and RC. Functions and algorithms on the radar control SBC is far less demanding, solutions shall consider this SBC if there is time.

This thesis aims to evaluate new architectures and solutions that can possibly replace current one when the time comes.

1.2. Thesis specifications

This thesis is targeted to evaluate possible processing architectures to replace the current PowerPC processing board. Focus will be on the Signal Processing (SP) unit, though if possible, i.e. if there is time the Radar Control (RC) unit should be considered.

The thesis consists of several subtasks, these are the following.

- *System evaluation and analysis,* a compilation of parameters from the current system which will be of importance when identifying requirements for new system. This will consider architecture, instruction set, software characteristics, theoretical performance and actual performance.

- *Requirements specification,* from system evaluation and analysis step compile a set of requirements that considers new theoretical performance, new practical performance. Sub requirements as operating system, architecture instruction set, supported programming languages and intercommunication bandwidth. New solution should be able to handle at least doubled workload.

- *Survey and analysis of new processing solutions,* with new requirements specification previously compiled collect a set of architecture proposals which at least satisfies the requirements specification.

- *Real-time analysis,* with a set of benchmarks specially chosen to asses qualities of the evaluated system.

- *Documentation,* shall be done describing results and conclusions reached during the thesis project. Documentation shall clearly state what performance to be expected from different architectures, what functionality they provide and their pros and cons.

1.3. Thesis limitation

Thesis should only consider possible solutions and preferably suggest one that has the best overall performance, scalability, usability, maintainability and cost. Performance will be evaluated through a series of benchmarks assessing important characteristics of a DSP. Benchmarks will target overall throughput, throughput for different operations commonly used by High Resolution (HR) algorithm.

For embedded computers power consumption is always of concern, both because of limited resources but also because of heat dissipation. However this is something that not has been a subject during this thesis.

2. System

The radar is composed of three different computer subsystems, each has its own specific responsibility. These subsystems are together with other components combined to a complete system that is capable of controlling the radar, extracting targets and present data to the operator.

2.1. Overview

Each subsystem is built of several different computer boards, some with general-purpose processors and others with more specialised hardware such as FPGA's. Boards with general-purpose processors are usually Single Board Computers (SBC) which has all the utilities it need to operate as a computer on one single circuit board. Several of these boards are then placed in a rack and connected to a backplane where different SBCs with different responsibilities communicate and work together. The picture below shows how the three different subsystems interact.

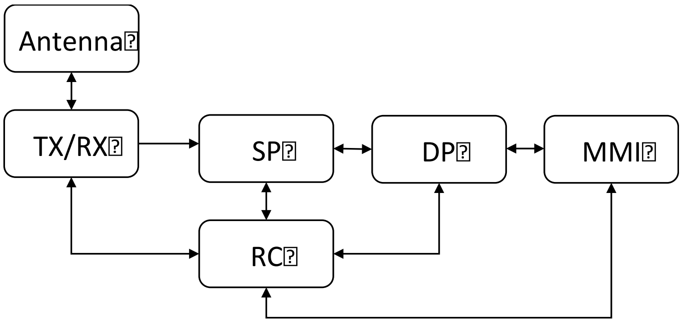


Figure 2.1. *Simplified chart subsystems SP,DP and RC*

SP, RC in *Figure 1* are built with one or several processor boards, FPGA-boards and interface-boards, these are then placed together in one rack while DP has its own. TX/RX is transmitter and receiver, the MMI is the user interface.

2.1.1. Signal Processor

The Signal Processor (SP) is the dedicated calculator of the radar. SP makes decisions about what is a possible target and what is not, with other words target detection. This is done by a series of calculations reaching from 1D max search to multiple dimension matrix operations and transforms. SP is usually used to perform large quantities of computations within a very small time frame. Most are typical linear algebra operations and are quite simple but carried out in big numbers, this is one of the more important topics for SP's.

The SP unit will deal with large quantities of data that would have no meaning for the interacting units, Data Processor (DP) and Radar Controller (RC), without SP's interpretation of the input data stream. After SP data will represent targets with position, direction and speed which are a fraction of the original data stream, thus putting less demand in throughput for DP.

SP is where the high-resolution algorithm runs.

2.1.2. Data Processor

The DP will deal with target plots that have been extracted by SP from earlier mentioned input stream data. DP handles mainly two different tasks, one is when the tracking algorithm matches new plots to old ones by prediction and comparison, the second is computing a map of the surrounding environment. The tracking algorithm will run continuously as a loop and will retrieve data from SP buffer. The second task is much heavier than the tracker, but it will run less often instead so demands are relatively low any way. If the radar is stationary it will only need to run once since it then has a map of the surrounding environment. If the radar is part of a moving unit it will have to update the surrounding environment with regular intervals to keep a fresh map of the environment.

2.1.3. Radar Controller

RC controls the entire system in hard real-time, which it does based on predefined settings from the operator and feedback from other subsystems. RC is in some senses much like the DP, it does not operate on raw input data but on data that has been abstracted by SP, thus does not put the same demands on high throughput. It can also be seen as a state holder, keeping track of settings manually set by an operator, such as Pulse Repetition Frequency (PRF) and Sample Frequency (SF). These settings will be used by both the SP and DP when interpreting the input stream and plot data.

3. Hardware

SP and RC are built with different configurations of boards. SP executes the HR algorithm on a DY4 board from Ixthos which later on became part of Curtiss Wright who renamed DY4 to CHAMP-AV. RC executes on a CPCI 6190 board by Emerson.

In this chapter different parameters of the boards and the overall hardware solution will be discussed.

3.1. Form factor

The current processing boards are built in Eurocard 6U form factor[1] [2], to minimize the impact for hardware migration it would be preferable to leave this unchanged. 6U is widely used and there exist a wide range of products to choose from.

A change would create a need for new rack, which in its turn would require every board in the rack to be changed as well.

3.2. Backplane

The SP/RC rack uses a hybrid backplane for communication within the rack. It is a hybrid of CompactPCI and VME, this makes it possible to build a rack with boards with different connecters but still makes it possible for them to communicate with each other.

Change of backplane would require some or all boards within the rack to be upgraded, causing more work. Thus it would be desirable to use VME and/or CPCI for future solutions as long as this does not have big impact on system performance.

CHAMP-AV connects to backplane via VME[1] and CPCI6190 uses CPCI [2].

3.3. PMC-GPIO

For the CHAMP-AV the data input stream is handled by a PMC module, the PMC module hosts a Virtex FPGA chip which receives data and then arranges data into suitable matrix representation before placing it in local SDRAM.

Input data from PMC is received via PCI interface [3].

3.4. Onboard Memory

For cPCI6190 has 512MB of onboard SDRAM ECC memory. The memory of the cPCI6190 is accesd on a 64-bit memory bus witch operates at 133MHz [2]. This would theoretically allow for data transfers at a rate of 1064 MB/s. This does not consider memory access strategy or address transfers so a this rate will most likely not be achieved in reality.

The Champ-AV has 128 MB SDRAM onboard accessible only to each cluster, and a global memory of 64MB accessible by both clusters and IOPlus. These memories work at 100MHz and can be accessed by a 64-bit wide PCI bus, however limited to 32-bit by SPE-PCI bridges which only work on 32-bit words. This results in bandwidth for data transfers at rates up to 400 MB/s. [1]

3.5. CHAMP-AV

For the heavier high-resolution algorithm that require higher throughput the CHAMP-AV DSP board is used. This is a quad processor board which hosts four Freescale MPC7410 with Altivec used as Signal Processing Elements (SPE's) and one MPC8240 CPE system controller. Below is the hardware specification of a CHAMP-AV DSP card [1].

Processors	4 x Motorola Power PC 7410 Altivec @ 500 MHz
Processors	1 x MPC8240 CPE @ 200 MHz
Global SDRAM	64 MB
Local SDRAM	128 MB
Level 2 Cache	2 MB
Flash Memory	8 MB
Backplane	VME

Table 3.1. *CHAMP-AV specification*

The CHAMP-AV has a setup of two SPE clusters, where each cluster has two MPC7410 SPE's working at a frequency of 500 MHz. Each cluster is connected to a SPE-PCI bridge which connects the clusters to a local 128MB SDRAM, a global memory of 64MB and flash, a PMC site local to each cluster and finally connects the clusters to a dispatcher which will handle assignment of tasks to each SPE. The dispatcher will also handle communication and initiate I/O with exterior units. Exterior communication is done through a General Purpose I/O PMC module which communicates with the backplane via a PCI interface, some communication is also done with Ethernet.

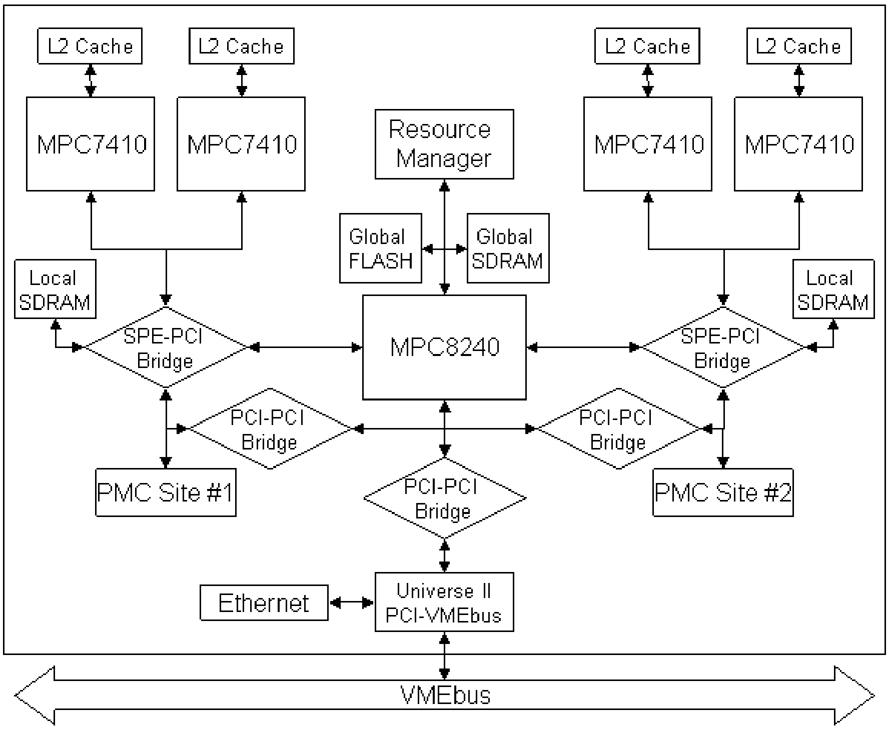


Figure 3.1. *CHAMP-AV block chart*

3.6. Power Core CompactPCI6190

For RC and less demanding tasks Power Core cPCI6190 processor board from Motorola is used. This is a single CPU board which provides less throughput but gains in simplicity, lower power consumption as well as lower cost. The CPCI 6190 board is built around the PowerPC 750GX CPU with a Discovery GT-64260 system controller. Below is a hardware specification. [2]

CPU	PowerPC 750GX @ 1GHz, 1MB L2 cache
CPU L2 cache	1 MB
Onboard Memory	512 MB ECC SDRAM
Flash Memory	64 MB
Backplane	Compact PCI

Table 3.2. *PowerCore cPCI6190 specification*

The CPCI 6190 is a PowerPC 750GX processor working at 1 GHz which provides high speed with low power consumption. For system control a Discovery GT64260 unit is used which connects the PowerPC 750GX with a 512MB high-speed SDRAM, Ethernet interfaces, flash memory and two PMC slots [2].

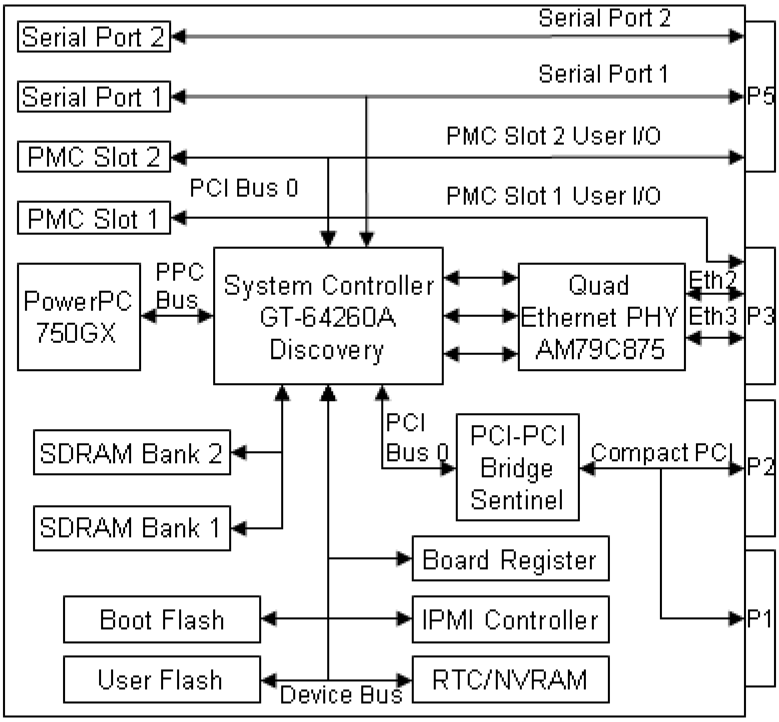


Figure 3.2. *PowerCore cPCI 6190 block chart.*

4. Software

To operate the platform an Operating System (OS) is required as well as Software Development Kit (SDK) for development of applications and proper software libraries which takes maximum advantage of underlying hardware.

4.1. Real-Time Operating Systems

Real-time operating systems also known as RTOS are operating systems that provides real-time functionality. There is different type of RTOS's, those that provide soft real-time functionality, and those that provides hard real-time functionality. There are also those that are native real-time capable and those that are not which depend on other modules or extensions to provide necessary real-time functionality.

RTOS's for soft real-time systems are those used for applications where lateness does not directly make result useless or imply catastrophic consequences. Instead consequences increase with lateness, which is accepted until some threshold is passed. RTOS's for hard real-time systems are those used for applications where lateness immediately makes results useless or causes some kind of unrecoverable loss or damage.

Hard RTOS's should be completely deterministic which makes it possible for developers to successfully predict real-time properties making it possible to avoid any unwanted behaviour.

Whether a system is soft or hard real-time will have to be evaluated for each system, even though a system might strike one as soft there might exist some certain circumstances which makes it hard. For this project we will be looking at hard RTOS's.

4.1.1. Present RTOS

For the lifetime of present solution a combination of VxWorks 5.4 and VxWorks 5.5 has been used. VxWorks is discussed next.

4.1.2. Wind River VxWorks

Among all RTOS's VxWorks is the most used and commercially successful. It has been deployed in many mission critical systems such as space crafts, aeroplanes and financial systems.

VxWorks has been built from scratch to host real-time applications, this is one of the big differences between VxWorks and real-time Linux distributions. VxWorks is highly deterministic and amongst other real-time capabilities it implements mutexs' and semaphores with priority inheritance to prevent priority inversion.

VxWorks is optimised to be deterministic, low latency and highly reliable. [4]

4.1.3. Montavista Linux

Montavista Linux is a native RTOS, claimed to be hard real-time. It is built with a pre-emptible Linux kernel which has been modified to be fully pre-emptible. The mainstream Linux kernel is only partially pre-emptible and Montavista provided this functionality, which then has been extended to be fully pre-emptible for Montavista Linux.

Montavista has been built to handle known real-time issues such shared resources and priority inversion amongst others [5].

4.1.4. Wind River Real-Time Core for Linux

Wind River points out that the Linux kernel does not deliver guaranteed real-time. The reason for this is that real-time mechanisms wasn't implemented from the beginning and Linux size makes it very hard to check all possible traces. This might cause real-time tasks to be blocked due to execution within unforeseen kernel non pre-emptible code.

To overcome this issue, Wind River has developed a real-time core which works together with the Linux kernel. The real-time core runs below the Linux kernel and executes non real-time Linux tasks with lowest priority, while all real-time tasks are loaded directly into the real-time core. Interrupts are initially handled by the real-time core which then routes them to the Linux kernel when no real-time tasks are waiting to be executed.[6]

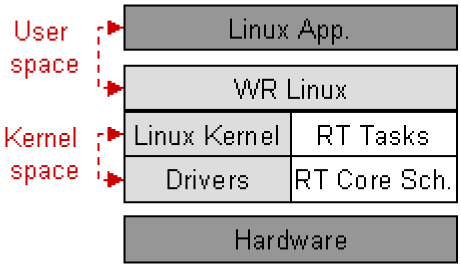


Figure 4.1. *Wind River Real-Time core hierarchy [7].*

4.1.5. Xenomai

Xenomai is a non-proprietary real-time patch, which enables Linux to handle hard real-time constraints in the same way as Wind River Real-Time core. It is a real-time core, which cooperates with the Linux kernel to handle hard real-time tasks.

Xenomai provides API's to proprietary RTOS's making migration significantly easier by mapping legacy RTOS operations to Xenomai equivalents. [8]

4.1.6. RTAI - Real-time Application Interface

RTAI is a Linux patch just as Xenomai, it provides a real-time core which enables Linux to handle hard real-time. As for Xenomai and Wind River Real-Time Linux, RTAI also uses a real-time core approach, which handles real-time tasks. RTAI also has a skin allowing legacy code to run on the RTAI kernel with little or no modification.

One difference between RTAI and Xenomai is that RTAI has put more effort to provide low latencies while Xenomai has put more work into making it portable.[9]

4.2. Operating system types

There are different ways of operating multi-core and multi-processors systems.

4.2.1. SMP - Symmetric Multiprocessing

Symmetric Multiprocessing can be used in a system where each core runs the same application. SMP type of OS's makes it easier to utilise all processor resources in multi-processor systems and multi-core processors. SMP runs as one single OS which controls all of the cores within the system, all cores are symmetric both from HW and SW point of view. This means that each core on the processor or board is as suitable as any other to handle a task or thread allowing simple dispatching and scheduling.

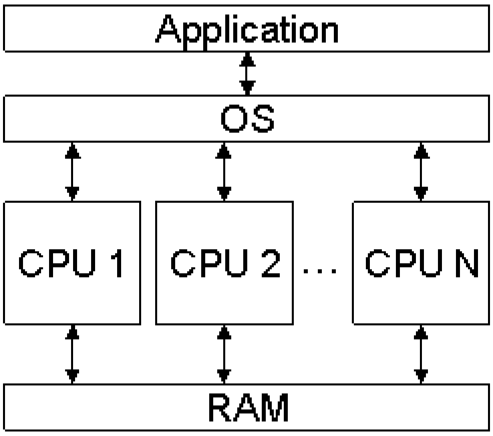


Figure 4.2. *SMP configuration.*

SMP makes it possible to build a applications which before was written to execute standalone, as one single application making system architecting much easier and software more adaptable to increasing number of cores on multi-core CPU's.[10]

4.2.2. AMP - Asymmetric Multiprocessing

Asymmetric Multiprocessing is the opposite of SMP. AMP systems run several independent processors and tasks. This makes it possible to add higher diversity within both hardware and software but with a higher complexity. This could be suitable if here is a need for very specialised hardware or software which need to run independent of the rest of the system.

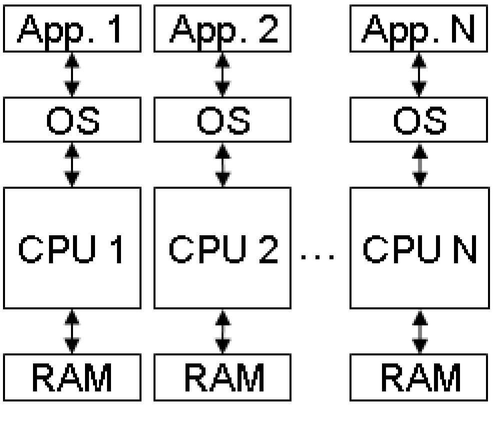


Figure 4.3. *ASMP configuration*

In this way a customised platform can be built on with each processor handling it's specific type of tasks. [11]

4.3. Additional Software

In addition to OS, software such as function libraries are strongly recommended for easier and more effective development.

4.3.1. Vector and DSP libraries

To make full use of the floating point performance, the operations need to be optimised for the underlying hardware. If SIMD architectures are used, the regular compiler will not be able to optimise binary to the SIMD instructions without manually coding these instructions. This requires specially written operations to be written for the available instruction set. This can either be done in-house by developers or in many cases there are proprietary libraries from vendors which have been highly optimised for best performance and portability.

The HR algorithm is currently implemented with the Ixthos IXLibs-AV DSP library, IXLibs is optimised for Altivec and the MPC74xx processors. IXLibs-AV provides a variety of important DSP operations such as FFT and many others.

There are several DSP/vector libraries available and in many cases the choice of hardware vendor will also mean that their libraries optimised for there hardware has to be used. Examples are GE-IP's AXISLib-x86[12] or Curtiss-Wrights Continuum[13].

4.3.2. VSIPL - Vector Signal Image Processing Library

VSPIL is a standard vector arithmetic's API [14], using VSIPL funtions and a VSIPL ported DSP library will maximise portability by allowing software to use different DSP libraries. Both Continuum and AXISLib comes ported to the VSPIL API.

5. High-Resolution Algorithm

The radar runs in two main modes, one mode which covers the entire plot area and a one second mode which allows the operator to choose up to four sub areas where to increase resolution, a high-resolution area. The high-resolution area is a fine grained area which makes it possible to separate two or more targets that might appear as one in normal resolution.

The high resolution algorithm is built with three different parts, the Range Bin Selector (RBS) which selects bins to look at, the MUSIC algorithm which computes data to extract plots from and last the High Resolution Extractor extracts and correlates HRFC plots with FC plots.

5.1. Range Bin Selector

The RBS takes input data, locates potential plots and then performs pre-processing to make data suitable for further computations. The RBS software is implemented in a sequential manner, and because of this benefits from high clock frequency, fast buses, fast memories and effective pipelines, characteristics of general purpose CPU's. [15]

5.2. MUSIC - Multiple Signal Classification

MUSIC is a signal processing operation which extracts several different components from an input signal. A signal can usually be seen as the sum of several other signals building up the received one, assumed that the number of components is known in advanced MUSIC can detect these. By sending out signals (components) on multiple frequencies the HR unit will be able to detect target within the same bearing and distance, by estimation of direction of arrival and Doppler shift.[16]

The MUSIC algorithm is the core of the high-resolution unit, this is where FC plots are resolved into finer grained HRFC plots. MUSIC uses several typical complex signal processing operations for which many are of vector or matrix form. Typical for these are that there are few operations that are done many times, which can successfully be executed in parallel.

However the MUSIC software is not entirely programmed for parallel execution, one of the pre-processing steps which also happens to be computational heavy has been implemented sequentially. Except for this step the remaining computations have been, where suitable, implemented with operations from IXLibs-AV DSP Library, *see Appendix C.*

Typical demanding operations frequently used are the Singular Value Decomposition (sequential) and Fast Fourier Transformation (parallel). [17]

5.3. High Resolution Extractor

The HRE handles high resolution plots from the output of MUSIC. It merges plots that belong to the same target and then correlates the high resolution plots to normal plots. As the RBS software this module is also implemented sequentially, thereby will not be effected by any parallelism that the processing solution enables.[18]

6. Processing solution types

In this chapter different technologies that are related to this thesis will be described, some that are used, being considered or just for perspective.

6.1. Processor types

There are different types of processors which differ in how they work and for what they were manufactured for. Here some relevant types will be discussed.

6.1.1. General Purpose Processors

General purpose CPU's or GGP's are those that you find in common Personal Computers (PCs). This type of processor has been built to perform well on all type of instructions. GP-CPU's have delivered sufficient performance for applications used by home users up until recent years. This has mainly been possible due to high increase in clock frequency, larger and faster on chip cache memory, faster buses etc. These are points that have been characteristic for of CPU developments for many years now.

However, big increases in frequency have also had a significant impact one power consumption and heat dissipation which has caused researchers' to look towards new ways to efficiently increase performance.

Trends are moving towards processors with parallel architectures, mainly by providing several cores on one chip allowing higher throughput at lower frequencies, lowering power consumption and heat dissipation. With more threads executing in parallel at the same speed as before increases the possible throughput.

PC's are replacing our multimedia machines, with the high demands on throughput caused by new formats such as High Definition (HD) TV, video games and graphics. This has put new requirements on CPU's. Multimedia has much in common with signal processing. New demands on PC's through multimedia has made manufactures well aware of the possibilities that SIMD offers to general purpose CPU's.

This is a trend that can be observed through out the last decade where manufactures has steadily increased development and more frequently implemented SIMD technology in their products. This has resulted in that there today is a higher diversity of SIMD technologies than before, for which many are equivalent or better than Altivec.

6.1.2. Graphic Processors

Graphics processors allow tremendous increase in throughput with its highly parallelised architecture. GPU's differs from regular CPU in the way that they are specialised to perform specific operations. Optimising each core to perform very well on a subset of the instructions supported by CPU's allows these instructions to be executed more efficiently. With a smaller subset of instructions each core can be made on a smaller die area and with lower heat dissipation. This allows for more cores per chip, thus providing a high level of parallelism.

GPU's can deliver a high throughput on a subset of applications and algorithms which benefits from this level of parallelism.

6.1.3. Digital Signal Processors

Digital Signal Processors more commonly referred to as DSP’s are constructed with the same idea as GPU’s, but instead specialised to do well on a subset of operations commonly used for signal processing. These type of processors are commonly specialised to perform FFT’s, filtering etc. The architecture allows these types of applications to greatly accelerate in performance while other algorithms and applications might not work particularly well.

As with GPU’s these type of processors suffer from being to specialised. They do not adapt well to applications that require more general type of computations

6.2. Execution models

There are several types of ways to execute instructions and computations here some relevant variations are touched upon.

6.2.1. SISD - Single Instruction Single Data

Earlier common CPU’s where of SISD type [19], for each instruction one pair of operands was used. I.e. the maximum throughput is equal to one operation per cycle.

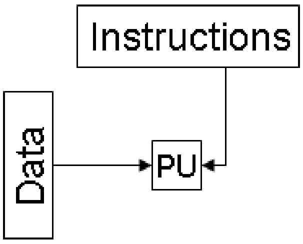


Figure 6.1. *SISD Model*

With the increasing role of instruction and data level parallelism these processors are becoming less and less important, and in many cases coexist with other types of execution models.

6.2.2. SIMD - Single Instruction Multiple Data

The answer to the problems related to GPU’s and in some extent DSP’s is Single Instruction Multiple Data (SIMD) [20]. SIMD is a technology that works as GPU’s but in smaller scale, it integrates a vector arithmetic unit into a general problem CPU. SIMD takes the better of two worlds, a general problem CPU that can effectively solve general problems while still accelerating its throughput for algorithms with higher computational intensity.

SIMD is a technology used to increase efficiency for some subset of operations carried out by processors. The use of SIMD technology can increase performance multiple times for some operations, commonly for arithmetic operations.

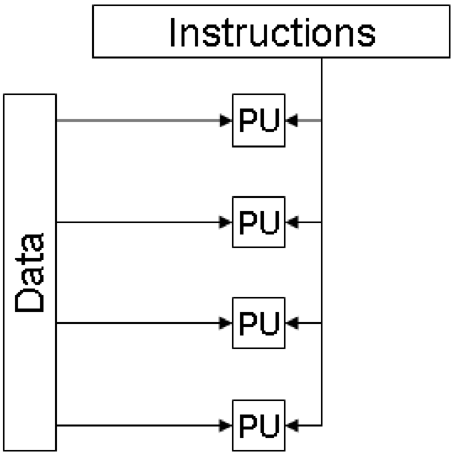


Figure 6.2 . *SIMD Model*

By executing the same instruction once on several data operands SIMD introduces a sense of parallelism. For operations that are used frequently and usually on large sets of data such as vectors or matrices performance benefits greatly compared to SISD where one instruction is executed for every single set of operands. Intuitively executing an instruction once on multiple operands instead of with multiple instruction calls can theoretically increase performance by a factor equal number of operands.

Commonly SIMD was only implemented on processors that were used in specific applications such as multimedia, radar and other types signal processing.

Below is an example with two vectors A and B, each with four operands. One instruction call with vectors A and B as operands will then have a throughput of four operations per cycle and instruction.

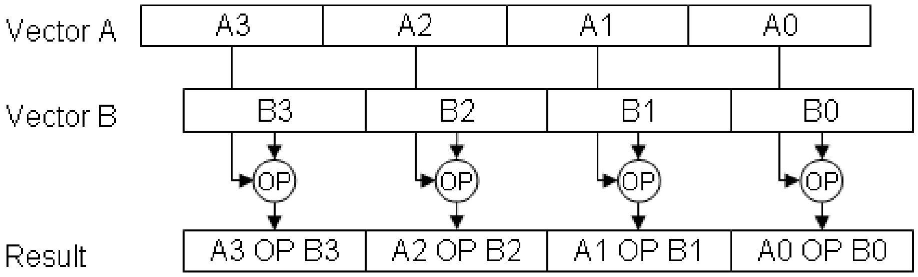


Figure 6.3. *SIMD example*

SIMD suffers though from some shortcomings, one is that manufacturers have their own technology and which has its own set of optimised instructions. This causes problem for software developers since the transition between different products might require developers to reconstruct there algorithms and software to the SIMD instruction set.

SIMD also set some requirements on hardware, such as special vector registers to handle vector operands and a special arithmetic unit to handle vector instructions.

6.2.3. MIMD - Multiple Instruction Multiple Data

Multiple Instruction, Multiple Data (MIMD) [21], more known as Multiprocessing. MIMD is the technology behind multiprocessing and multi-core technology. MIMD architectures make it possible to work on several data streams at once with different instructions.

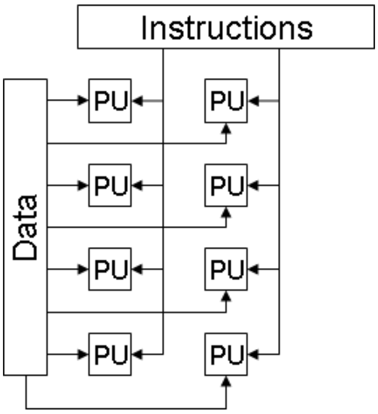


Figure 5.4. *MIMD model.*

6.2.4. SIMD or MIMD

Multiprocessing allows more tasks to execute in parallel, making it possible with higher throughput. All problems are not suitable for multiprocessing, problems may not be divisible and spread over multiple cores because of dependencies. This constrains the possible performance increase for multi-core technology, this is also known as Amdahl’s law [22].

Some tasks will always perform better sequentially thus constraining the utilisation level for multiple cores. This might be the case if two tasks work on the same set of data, to shuffle the data back and forth between the two cores would be inefficient.

SIMD on the other hand can not handle the separation of tasks but instead it can accelerate the performance on a single core. A task or thread that performs a large numbers of the same operations and on multiple streams of data will if optimised properly improve its performance by a factor equal to the number of parallel computations. As for Altivec which uses 128-bit vector operations, e.g. four 32-bit floating point operations in the same cycle compared to one in a SISD unit.

Performance will be strongly coupled to the type of applications running on a specific type of architecture. To achieve best possible performance, a architecture should preferably be chosen after what application it is supposed to handle. In some cases an architecture composed of several of the mentioned technologies will be the best solution while others may benefit of a architecture based on just one of the mentioned processor types.

Which one to use mostly depends on the application, trends are a combination of both, multi-core processors with a SIMD unit for each core. This makes it possible to get both a high level of data and instruction parallelism.

7. Benchmarking

To compare different architectures some metric is required. As a metric I have chosen to compare execution time for the old platform and then for one of the proposed new boards. Two different benchmarks has been done, both of importance to the HR algorithm and

7.1. Singular Value Decomposition – SVD

"Singular value decomposition is an important factorization of a rectangular real or complex matrix, with many applications in signal processing and statistics. Applications which employ the SVD include computing the pseudoinverse, least squares fitting of data, matrix approximation, and determining the rank, range and null space of a matrix." [23]

The SVD retrieves the Eigen values and Eigen vectors from the input matrix, the eigen values are then used to extract plots through a series of computations.

This method is implemented entirely sequential, it runs from beginning to end without the usage of any SIMD instructions. I have used a modified version of the SVD benchmark from the HPEC Challenge Benchmark suite [24].

The SVD benchmark runs four different test cases, common for all of them is that each window is maximised. That means that each bin within the window is passed to the *SVD* method, this represents the case where a target is detected for each bin.

Each variation of the benchmark runs one to four input windows, four is the maximum number of high resolution windows.

Since *SVD* is sequential this benchmark will give a good measure of the non-SIMD floating point performance.

7.2. Fast Fourier Transform – FFT

"A fast Fourier transform is an efficient algorithm to compute the discrete Fourier transform (DFT) and its inverse. A DFT decomposes a sequence of values into components of different frequencies. This operation is useful in many fields (see discrete Fourier transform for properties and applications of the transform) but computing it directly from the definition is often too slow to be practical. An FFT is a way to compute the same result more quickly: computing a DFT of N points in the naive way, using the definition, takes $O(N^2)$ arithmetical operations, while an FFT can compute the same result in only O(N log N) operations."[25]

FFT's works as a base for several signal processing operations. It transforms discrete samples of a signal into the frequency domain, which then can be used to perform useful operations for target detection.

FFT's depends heavily on Multiply Accumulates (MAC) [26], which SIMD execute very efficiently. MAC's are a operation which multiplies two values and then adds another, Because of the importance of FFTs and its use of the SIMD optimised DSP libraries FFT's have been chosen to evaluate the SIMD floating point performance as well as for giving a picture of the DSP libraries.

8. Analysis

Analysis will be made of both hardware and software algorithms. Because of the time constraints of this project and the amount of work related to actually benchmarking all hardware with optimised software, much of the numbers will be based on theoretical values. How a specific type of algorithm or operation maps to different architectures and instruction set will be theoretically evaluated and briefly tested in real-time at a high abstraction layer.

8.1. Processors

I have chosen to look closer at three different processors, the decision has been made on availability of rugged SBC with these processors, there peak performance and how well current software can map to the new HW platform. The table below summarises current and new processors.

Parameters	Freecale MPC750GX	Freescale MPC7410	Freescale MPC7448	Intel Core i7-610E	Intel Core i7-2715QE
SBC's	CPCI6190	CHAMP-AV	CHAMP-AV4	VR12, CHAMP-AV5	XVR14, CAHMP-AV8
GFLOPS	1 GFLOP	4 GFLOP	10 GFLOP	40.5 GFLOP	134.5 GFLOP
Cores	1	1	1	2	4
Frequency	1 GHz	500 MHz	1.25 GHz	2.53 GHz	2.1 GHz
Mem. Band.	-	528 MB/s	2 GB/s	17.1 GB/s	21 GB/s
L1 (Cache)	32 KB I-cache 32 KB D-cache	32 KB I-cache 32 KB D-cache	32 KB I-cache 32 KB D-cache	32 KB I-cache/c 32 KB D-cache/c	32 KB I-cache/c 32 KB D-cache/c
L2 (Cache)	2 MB	2 MB	1 MB	256 KB/core	256 KB/core
L3 (Cache)	-	-	-	4 MB	6 MB
Inst. Exten.	-	Altivec	Altivec	SSE 4.2	SSE 4.2, AVX
FP Perf.	1 GFLOP	4 GFLOPS	10 GFLOPS	41 GFLOPS	134.5 GFLOPS
TPD	-	6.3 W	8.5 W	45 W	45 W

Table 8.1. *Old and new processors.*

8.2. SIMD

In this section different SIMD vector instruction sets are discussed.

8.2.1. Altivec

Altivec is a vector instruction set and its supporting hardware. Altivec enables the programmer to execute up to four floating point operations with just one instruction. This allows Altivec enabled architectures to increase its throughput many fold for applications that requires a big number of repeated instructions. Altivec uses 128-bit vector registers to store its operands, when a instruction is issued it is applied to each word in the vector registers. This allows the processor to carry out what would have been four instructions on a Single Instruction Single Data (SISD) processor in just one instruction call.

When issuing an Altivec instruction that makes use of its parallelism the processors can load four 32-bit words into a 128-bit vector register. When all the operands has been loaded it issues the instruction which will apply to each of the elements, floating point words, in the input vector registers.[27]

8.2.2. SSE - Streaming SIMD Extension

Streaming SIMD Extensions (SSE) is Intel’s equivalent to the Altivec SIMD instruction set. SSE has gone through a series of refinements and is currently at version SSE 4.2 [28]. The current version uses 128-bit vector registers which supports instructions on up to four 32-bit single precision floating point numbers, real or complex. On a clock to clock basis, SSE 4.2 and Altivec provides equal FLOPS. However the Core i7 processors works at much higher frequencies compared to high-end PPC’s with Altivec support.

8.2.3. Advanced Vector Extension

Advanced Vector Extension (AVX) [29] is Intel’s latest addition to the vector instruction set. AVX has been extended to handle 256-bit vector arithmetic that is, eight single precision floating point numbers. This gives a doubled clock to clock FLOP performance compared to SSE 4.2 and Altivec. AVX combined with Intel’s raw performance enables exceptional floating point performance for general purpose processors.

8.3. Subsystem and software architecture

Here different components and subsystems are discussed.

8.3.1. Data Processing - DP

Through interviews with personal who manage the DP system I reached the conclusion that the DP is much less demanding in special functionality. Its primary task that has hard real-time requirements is the tracking that will have to, for all plots, search through old plots and find a match before next set of plots are retrieved. The mapping of the surrounding environment has less real-time constraints, since it does not execute very often, minutes or even tenths of minutes it has more than enough time to execute thus no hard requirements of execution. With more high-resolution areas we can expect that we will get more plots than before. The increase in workload can be expected to be in the range n^2, n is number of targets.

According to earlier measurements it has been determined that most of time for execution of the tracking algorithm is used for memory handling, which is moving and shuffling memory around.

DP does not require the same type of special instructions as SP, since there is no need to process large quantities of data but instead searches a set of plots. Higher performance for this unit can instead be increased by fast memories, bigger caches and better memory strategies.

8.3.2. Radar Control - RC

The RC unit will keep track of current settings for different parts of the radar. These setting will be compiled to one header which are sent out to each unit requiring information. The header is the same for each unit, each specific unit will then extract relevant parameters from the header.

RC does not require high throughput but instead high speed, this because it spends most of its time extracting, checking and compiling headers where no advanced operations will be carried out.

Through analysis by Saab of the RC the conclusion has been reached that RC is not able to process data properly when sending and receiving at speeds higher than 4-4.5 kHz. With faster processors, operating at higher frequencies this can possibly be solved.

The kernel of current VxWorks version is not up to date, a upgrade or change of RTOS can possibly address these issues.

8.3.3. Signal Processing and HRFC platform

Results from tests already made by Saab show that the MUSIC software has the highest demands, workloads that are extreme within this application segment makes this clear. Summary of test results below.

RBS megatest	69 ms
MUSIC megatest	3.45 s
HRE megatest	206 ms

Table 8.2. *Summary of HR SWU tests [30].*

These results show very clearly that the “MUSIC megatests” execution time is of magnitude much greater than those of others relevant SW units. When compared the MUSIC megatest execution time is at least 16 times greater.

This is useful because we know that most of the operations within the MUSIC SWU are parallelised with Altivec instructions.

Since each MUSIC computation should preferably run as a single thread/core[17], it is important to get higher performance per core. We also know that MUSIC spends much of its execution time on FFT’s and other operations that can be can be handled efficiently with vector arithmetic’s. Two important components for higher performance for this application are CPU speed and vector floating point arithmetic performance.

Only by looking at the CPU frequency we can see that, on general problems (non DSP) the Intel will most likely out-perform the Freescale.

Numbers indicate that test result will reveal a significant increase in both raw performance and in FLOPS.

8.3.4. Video Handler – I/O

The Video Handler is hosted on a FPGA chip on a PMC module, which receives data via backplane. When data has been received it arranges data into matrix form which signal processing later is performed on.

This can be kept as it is or it can be removed introducing two new options. One alternative I/O solution would be to let the Video Multiplexer do the pre-processing of data before sending it directly to IOP on HRFC board. A second alternative would be to let the IOP receive unarranged data and let it handle it do the pre-processing by itself.

8.3.5. Load balancing for multi-board solutions

For multi-board solutions there is a need to balance the workload between multiple boards, one possible way of handling this is letting the Video Multiplexer address to which board or CPU within a cluster that data shall be sent. A simple round-robin scheduler would be sufficient. Exact how to implement this will depend on what SBC is used.

8.3.6. Radar input data

There are several alternatives depending on which board and which type of backplane that it uses. Common for the majority of relevant boards is 4x-8x PCIe with bandwidth ranging between 1GB/s and 4GB/s in each direction. The PCIe interfaces are commonly connected via PMC/XMC-modules.

8.3.7. Multi-core and SPE tasks

Present implementation uses a parallel architecture where each SPE task is allowed to run undisturbed on its own physical processor. SPE's will execute from beginning to end without any interaction with surrounding environment for the signal processing part.

Because each SPE works independently of other SPE's, throughput will be able to increase close to linearly. However the increase in throughput will be constrained by the IOP since it will have to set up data and notify when new data is available to SPE's.

Throughput and performance will also benefit from higher clock frequencies since this in combination with modern memories, busses and cache strategies will increase the amount of data that can be provided and processed in some time frame.

Looking back at *Table 8.1* it is easy to see that the memory bandwidth has increased with a factor larger than that for increased peak FLOPS for any processor compared to the MPC7410. Because performance increase in memory bandwidth has a made the gap between processing capacity and memory bandwidth smaller a higher fraction of the processing capacity can be utilised.

More cores will increase the number windows that can be handled while higher frequencies and modern architectures will make it possible to handle more HR areas or compensate for fewer cores. Intuitively a combination of multi-core and fast processors will be ideal from a performance point of view.

8.3.8. RTOS

The article [33] from 2008 a comparison has been made between VxWorks, Linux, RTAI and Xenomai which outcome showed that foremost Linux with RTAI delivers performance very close to that of VxWorks. Xenomai and Linux show convincing results, however since Linux kernel is not hard real-time capable it is not advised to use it for this kind of systems.

With the results for Linux and considering that Montavista Linux is hard real-time capable, MV Linux is a possible alternative to the mainstream Linux kernel but for hard real-time.

9. Processing solutions

All of the suggested Intel platforms will require software changes to FC/HRFC because of the migration from Altivec. Some SWU's will require redesign and reimplementation depending on the suggested platform. Following processors are used for different alternatives.

Parameters	**Freescale MPC7410**	**Freescale MPC7448**	**Intel Core i7-610E**	**Intel Core i7-2715QE**
Board (Alternative)	Present	CHAMP-AV4 (A1)	VR12 (A2, A3), CHAMP-AV5 (A3)	XVR14 (A4, A5), CHAMP-AV8 (A5)
SBC's	CHAMP-AV	CHAMP-AV4	VR12, CHAMP-AV5	XVR14, CAHMP-AV8
GFLOPS	4 GFLOP	10 GFLOP	40.5 GFLOP	134.5 GFLOP
Cores	1	1	2	4
Frequency	500 MHz	1.25 GHz	2.53 GHz	2.1 GHz
Mem. Band.	528 MB/s	2 GB/s	17.1 GB/s	21 GB/s
L1 (Cache)	32 KB I-cache 32 KB D-cache	32 KB I-cache 32 KB D-cache	32 KB I-cache/c 32 KB D-cache/c	32 KB I-cache/c 32 KB D-cache/c
L2 (Cache)	2 MB	1 MB	256 KB/core	256 KB/core
L3 (Cache)	-	-	4 MB	6 MB
Inst. Exten.	Altivec	Altivec	SSE 4.2	SSE 4.2, AVX
FP Perf.	4 GFLOPS	10 GFLOPS	41 GFLOPS	134.5 GFLOPS
TPD	6.3 W	8.5 W	45 W	45 W

Table 9.1. *Processors used in different SBC alternatives.*

9.1. Building blocks

The HRFC platform is built of several different components. Some are dedicated for signal processing while other filter and manages data to make it available to the signal processing.

- Video Multiplexer [34], video multiplexer transfers filtered data to processing board.
- Video Handler [34], receives data, arranges it into matrix form and then stores it locally until requested.
- IOP [34], handles I/O for signal processing and communicates with DP.
- SPE [34], signal processing elements which are dedicated to only handling signal processing.

9.2. Current platform

Current HR platform uses a quad MPC7410 from Curtis-Wright. It has one IOP which runs on a MPC8240, and two SPE clusters, each with two MPC7410 SPE's theoretically capable of handling 4 GFLOP's. This gives the board a total of 16 GFLOPs dedicated signal processing capacity.

Data is received from backplane which is handled by the PMC-GPIO module, it arranges received data and stores it in a local memory.

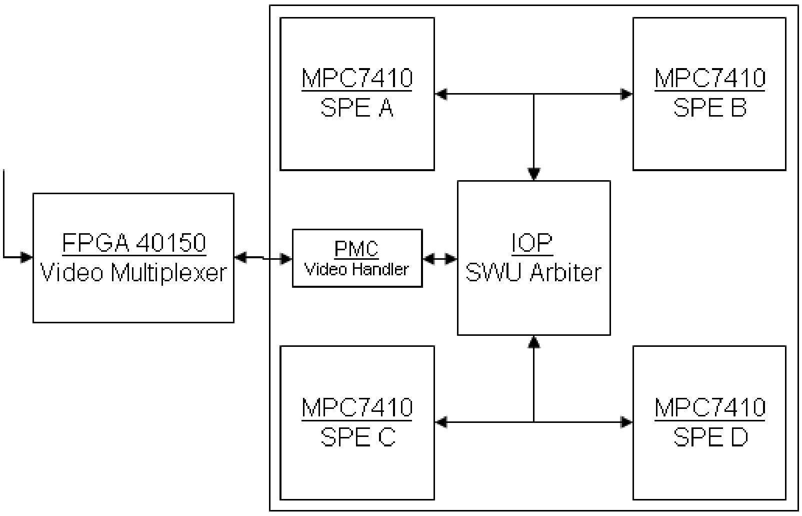


Figure 9.1. *CHAMP-AV, MPC7410 SBC component mapping.*

9.3. Alternative 1: Quad PowerPC 7448

It is possible to use a quad PPC7448 board. This option has the benefit that software could be moved to the new board with minimal changes. MPC7448 has the same architecture as the MPC7410 which is the processor on the DY4 board.

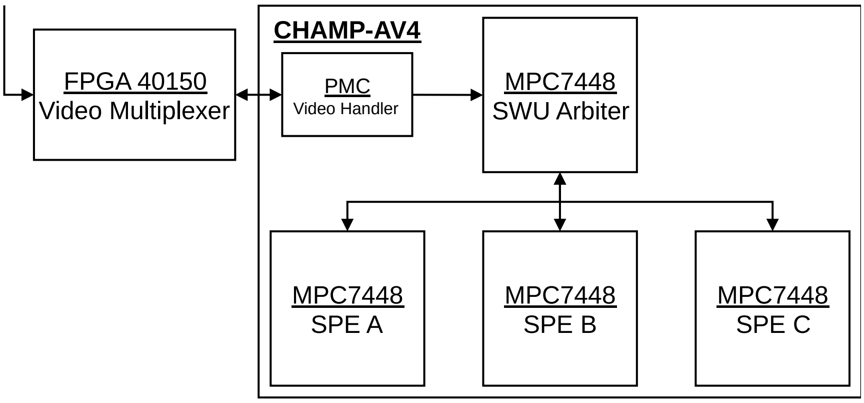


***Figure 9.2**. Quad MPC7448 SBC architecture.*

9.3.1. Expected performance

The PPC7448 processor is built with the same core as PPC7410, software moved to the 7448 would run seamlessly. The difference between the two processors is that the later runs at a higher frequency, 1.25 GHz compared to 500 MHz. The total computational capacity is 4 GFLOPS, that is 10 GFLOPS per processor.

Figure below gives a closer look at how software units and components can be mapped with minimum impact and maximum performance.

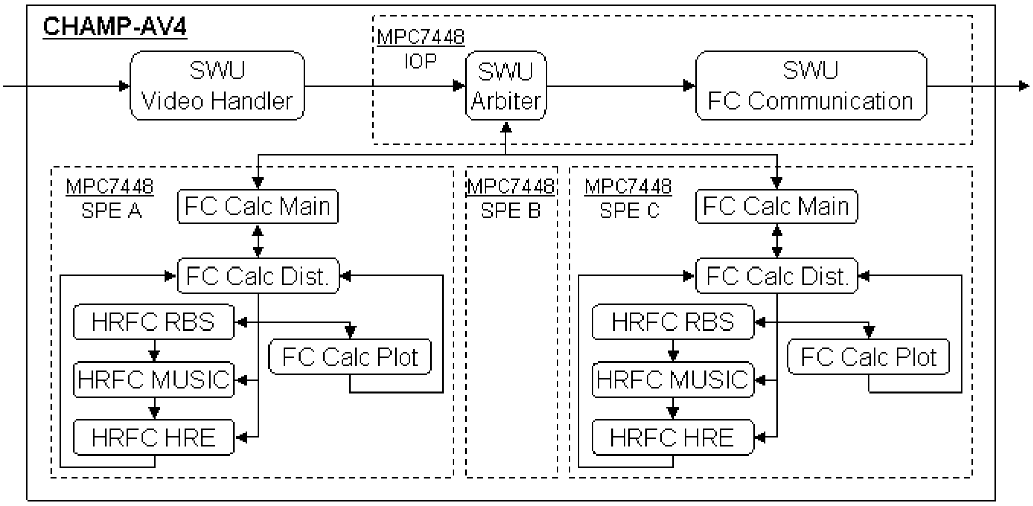


***Figure 9.3**. Quad MPC7448 SBC SW mapping.*

Since this board is built without a designated IOP processor, one of the four MPC7448 will have to fill this place. This would result in a board which host three SPE's and one IOP, the SPE's and IOP could possibly communicate through the Gbe switch which can provide a one to one communication between any of the processors.

Three MPC7448 SPE's are capable of handling 30 GFLOPS compared to four MPC7410 which are capable of 16 GFLOPS, the increase in performance with a new quad MPC7448 solution with three SPE's would then be about 88% better than the present solution.

9.4. Alternative 2: Single Core i7 SBC

The single *Core i7-610E* boards provides both performance and low power consumption. Intel has limited the power consumption to current level thus ensuring that power consumption stays low while performance increases with new generations of processors.

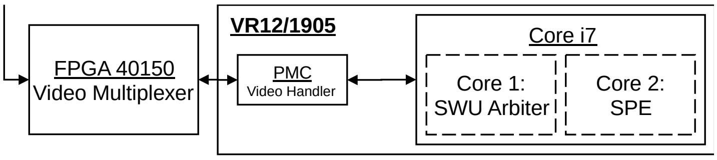


***Figure 9.4**. Single Core i7 SBC architecture.*

9.4.1. Expected performance

The Core i7 processor available on the embedded SBC's has a dual core. Since the considered cards do not have a separate IOP chip it will suitably be replaced by one of the processor cores. One core equals 50% of the total processing power, this leaves the second core and the other 50% to dedicated signal processing without any disturbance from interrupts that instead are handled by the "IOP".

One possible SW mapping would be to use one core to replace the "IOP" functionality, Arbiter etc, and one as SPE (signal processing).

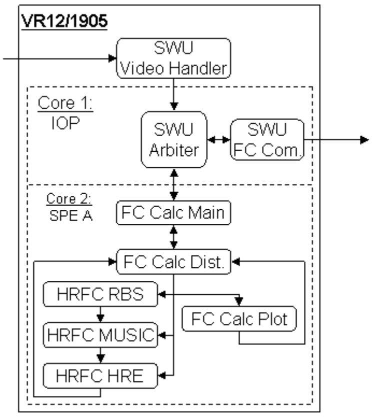


Figure 9.5. *Core i7 SBC SW mapping.*

The Core i7 board has a peak performance at ~41 GFLOPS, this break down to roughly 20 GFLOPS per core compared to the CHAMP-AV which peaks at 4 GFLOPS per core, assuming that SSE is used. Theoretically one Core i7 SPE could handle the load of five MPC7410 SPE's, the CHAMP-AV board has four MPC7410 SPE's. That is 20/16 or a 25% performance increase.

9.5. Alternative 3: 2 x Core i7 SBC / Dual Core i7 SBC

A second approach would be to use two Core i7 boards, or a dual Core i7 board. Each processor would be configured as for the single processor solution. This will have a significant impact on performance when compared to the DY4 solution with only little extra effort compared to the single Core i7 solution. The extra work required for a dual board or dual CPU solution is required for load balancing between the two (or more).

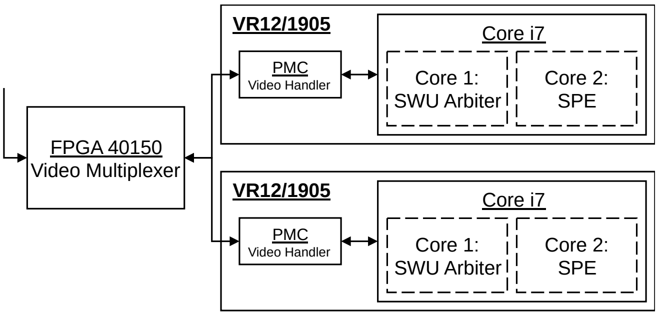


Figure 9.6. *2 x Core i7 SBC architecture.*

An alternative would be to use a single board dual CPU architecture, e.g. if there is not enough free slots in the backplane.

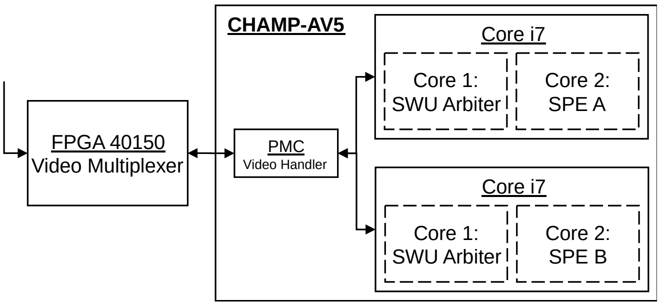


Figure 9.7. *Dual Core i7 SBC architecture.*

9.5.1. Expected performance

The performance of a multiple processor solution can be expected to scale to the number of processors, so a dual Core i7 solution would theoretically be able to handle twice the workload as the single processor solution, that is the same load as 10 SPE's implemented on ten MPC7410. That is 40.5 GFLOPs of processing performance or a 153% performance increase compared to the CHAMP-AV solution.

A closer look at how software would be mapped to the dual board alternative, shown below.

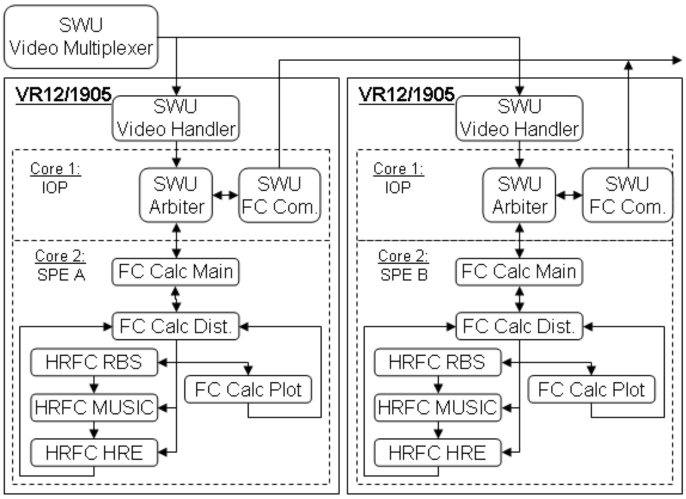


Figure 9.8. *2 x "Sandy Bridge" SBC SW mapping.*

For the single board dual CPU alternative, this is basically the same except with load-balancing done by Video Handler.

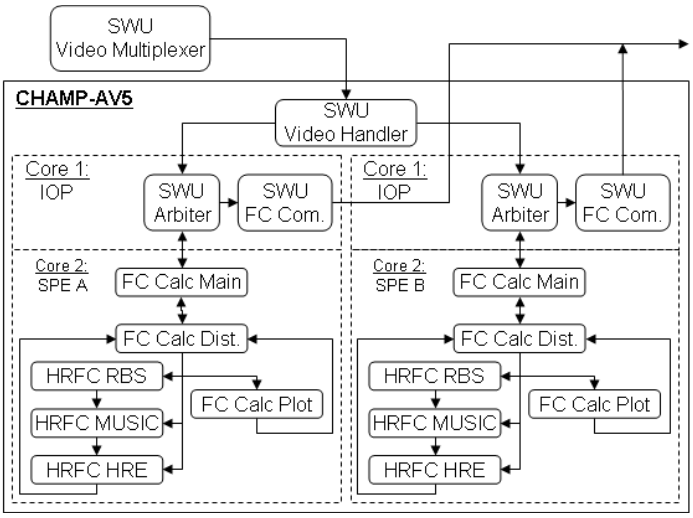


Figure 9.9. *Dual Sandy Bridge" SBC SW mapping.*

9.6. Alternative 4: Single 2nd Generation Core i7 (Sandy Bridge)

The 2nd Generation Core i7 boards differ form other processors in one significant way, which is it integrates the GPU on the CPU chip, called IGP. This allows the processor to make use of GPU type of computing with minimal transfer times and latencies. GPU's have been shown to perform very well for DSP applications.

However the IGP is not the only change, one that will have a big impact on DSP is the new AVX instruction set which uses 256-bit vector instructions instead of the 128-bit instruction set used with SSE4.2 and Altivec. AVX theoretically enables doubled performance for FP arithmetic.

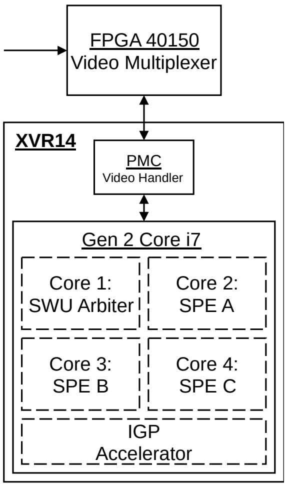


Figure 9.10. *2nd Generation Core i7 SBC architecture.*

9.6.1. Expected performance

The first available of these boards come in quad core versions, which will give us a solution with one IOP and three SPE's.

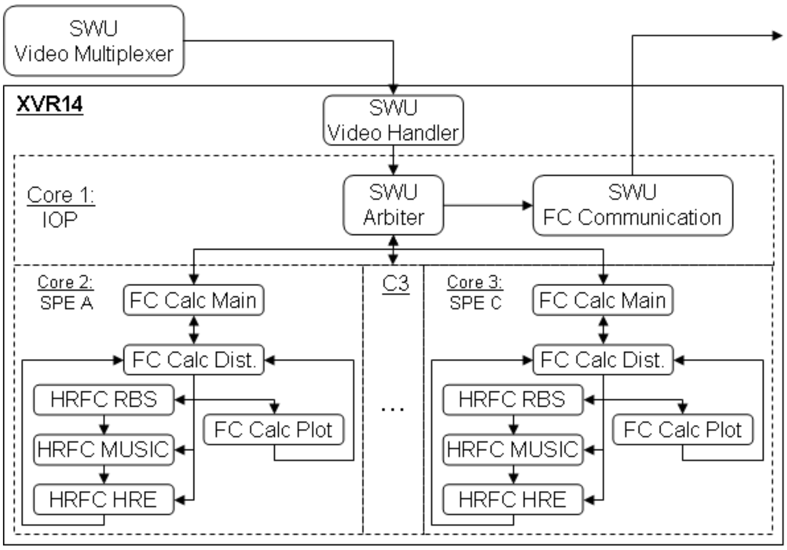


Figure 9.11. *2nd Generation Core i7 SBC SW mapping.*

This board is theoretically capable of 134.5 GFLOPs, 9.5 GFLOPS can be assigned to the IGP, this leaves 31.25 GFLOPS per core. This is more than seven times the processing performance compared to each SPE on the MPC7410. Assuming that one core is used as "IOP" as for the other Intel solutions. This leaves three cores to be dedicated to SP, if each core is capable of handling seven times the load as one MPC7410 this will result in a platform theoretically capable of handling twenty one times the workload as the CHAMP-AV board. That is 486% increase in floating point performance compared to the CHAMP-AV solution.

9.7. Alternative 5: 2 x Core i7 (Sandy Bridge) SBC/ Dual Core i7 (Sandy Bridge) SBC

A dual Sandy bridge solution would provide performance at rates close to those of GPU's, which will be mentioned next. However compared to GPU's the work related for migration will be less and the single core to single SPE model can be maintained. It can either be realised with a two single CPU boards or one dual CPU board.

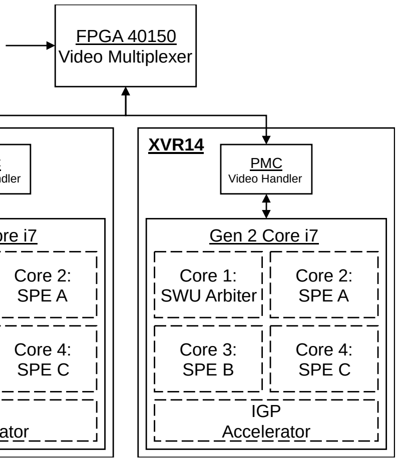


Figure 9.12. *2x 2nd Generation Core i7 SBC architecture.*

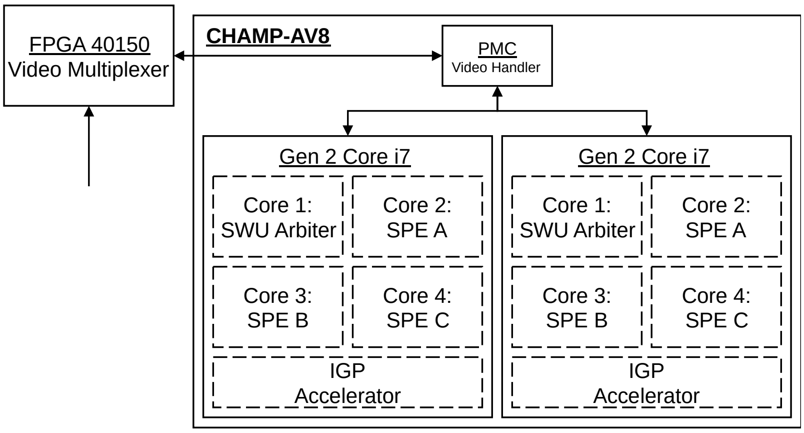


Figure 9.13. *Dual 2nd Generation Core i7 SBC architecture.*

9.7.1. Expected performance

A dual Sandy Bridge solution would theoretically deliver 269 GFLOPS of floating point performance. 9.5 of GFLOPS per CPU comes from the IGP, leaving 250 GFLOPS. This solution would dedicate six of the eight cores to signal processing, this is a 1072% increase in floating point performance compared to CHAMP-AV solution.

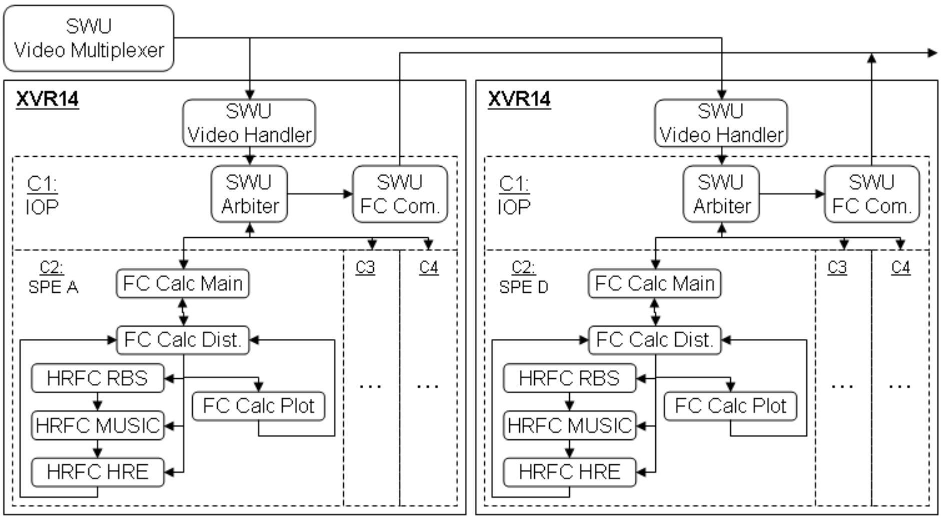


Figure 9.14. *2x 2nd Generation Core i7 SBC SW mapping.*

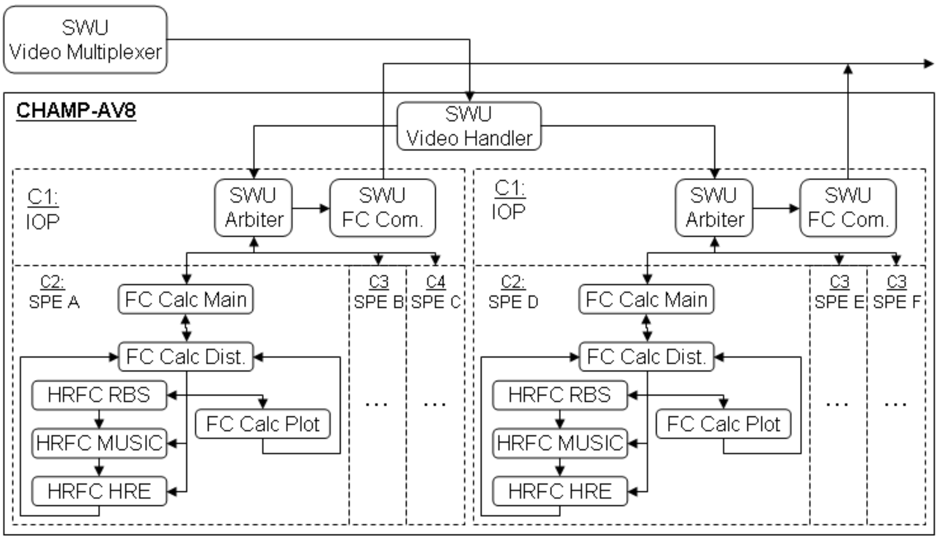


Figure 9.15. *Dual 2nd Generation Core i7 SBC SW mapping.*

Dual processors boards come for VPX as the single processor board, however the single processor board is available for VME.

9.8. Alternative 6: Core 2 Duo with GPU accelerator

One option is to use a General Purpose GPU's (GP-GPU's), with GP-GPU's it is possible to accelerate performance to extreme rates. These platforms uses a general purpose processor to handle the basics and then offloads the computation intensive task to the GPU.

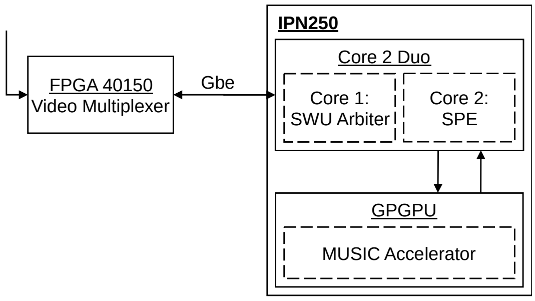


Figure 9.16. *Core 2 Dual with GP-GPU SBC architecture.*

9.8.1. Expected performance

A key issue with this solution is that it can not use the approach where each SPE work independently from top to bottom as for the other proposed solutions. Additionally the GPU is constructed to work well with a small subset of operations commonly used in graphics and those who are highly parallel, thus only some parts of the software is suitable to run on the GPU. Several SPE thread will have to run in parallel on the same hardware.

The MUSIC calculations which now use SIMD would be suitable to move to the GPU. The remaining more general and sequential SW would not benefit from the GPU or might even suffer from being executed on the GPU, thus non MUSIC related computations should not be re implemented for GPU.

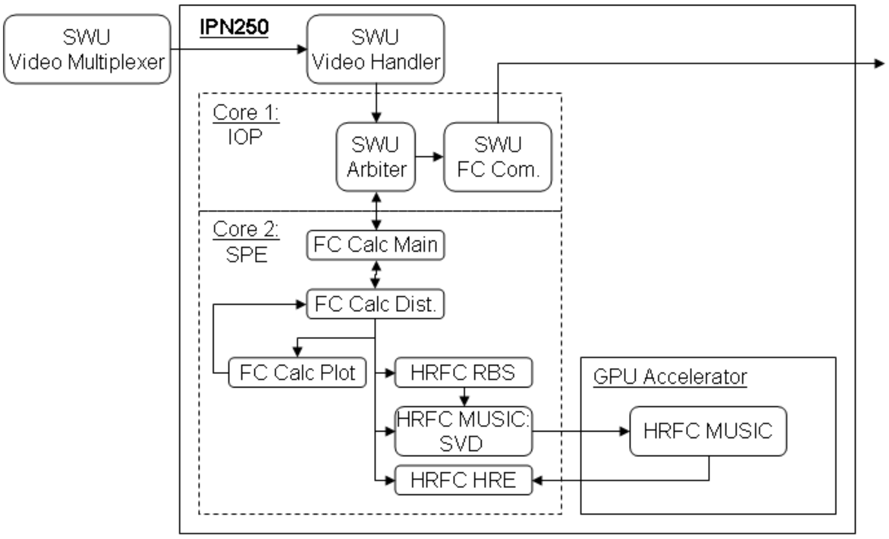


Figure 9.17. *Core 2 Dual with GP-GPU SW mapping.*

Estimation of performance for GPU's is much harder since much of the processing performance is kept within the GPU. The GPU has been optimised to perform calculations on relatively big sets of data compared to today's SIMD units. The cost of transferring data back and forth between the CPU and GPU might actually cost more than the computational gain.

Since the GPU solution requires algorithms to be re implemented in a way that takes advantage of the GPU’s strengths it is hard to compare the solutions without practical testing.

With only fragments of the SW running on the GPU and the fact that this is where most of the computational capacity is kept, the CPU might become a bottleneck. To truly take advantage of the GPU it is required by the CPU to manage the sequential parts of the SW, and the CPU will risk being overwhelmed by this task thus substantially lowering the GPU peak performance. Peak performance for this solution is 390 GFLOPS, but with a CPU/GPU ratio at 5/385. This may become a serious bottle neck unless most of the software is vectorised.

10. Migration

Implications caused by new solution.

10.1. PPC solution

The PPC solution connects to a VME backplane thus not introducing any implications to the rest of the system environment. PMC sites supports PCI-X which is backwards compatible with PCI, connection to backplane can be unchanged. If more bandwidth is desired the PCI-X interface could be used instead supporting up to 800 MB/s, this requiring some redesign of backplane I/O.

Software migration to a new PPC solution would result in the minimal possible re-design. Since the PPC solution runs on the same type of e600 core with Altivec as present solution SPE software will run on the MPC7448 without any redesign.

There are two sources of changes, first of all there is no dedicated IOP hardware which requires one of the MPC's to full fill this responsibility and second intercommunication can possibly require some redesign. This depending on the bandwidth required for intercommunication, there are three possibilities which are keep using PCI, upgrade to PCI-X or Gigabit Ethernet.

As for any of the different solutions the PPC alternative is scalable allowing more boards to share the work. This will require some load balancing function within the Video Multiplexer which sends data to the signal processing board.

10.2. Intel solution

Moving to a Intel platform will require more re-designing, a big advantage though is that all Intel variations require about the same level of migration work and about the same changes. When the migration has been carried out to say a Core i7 dual core with SSE 4.2 it will later be possible to move it to a Core i7 quad core with AVX. This is possible since DSP libraries from one vendor in most cases uses the same API for SIMD optimised operations.

The Intel solutions should be implemented so that each CPU works as a SMP cluster, compared to present board where each CPU is part of a SMP cluster. If a dual core is initially used it could seamlessly be moved to a quad core. This aspect of maintainability is very beneficial for future platform renewals, perhaps allowing more often platform upgrades but with minimal work.

Adapting software to a entirely new platform will require re-design, first of all software has to be adapted to a SMP operating environment which will requires some structural changes. Old DSP library specific functions will have to be changed to new DSP library methods, this however will be hard to avoid for any solution except Alternative 1, PPC platform.

There is no dedicated IOP hardware on these boards either, requiring one of the CPU cores in each SMP cluster to act as IOP. This can be done with some SMP OS's. For example, VxWorks 6.8 makes it possible to dedicate one core to one specific task.

There are several Intel options two single board, single CPU and two single board, dual CPU. Some with PCI-X, x8 PCIe Gen 2 and sRIO Gen 2, bandwidths range all between 800 MB/s to 8 GB/s. 8 GB/s is only possible for boards for VPX backplane where a sRIO interface exists directly towards the backplane providing 8 GB/s peak bandwidth. Depending on which interface to the backplane is chosen different amounts of work is related to I/O adaption.

As with the PPC solution there is a possibility to scale the number of boards depending on processing requirements, for all multi-board solutions the Video multiplexer is required to balance load. There are to variations of Dual boards, one with dual Core i7-610E and one with dual Core i7-2715QE. The first is connects to a VME slot and the second to VPX.

For single board, dual CPUs there are a few options when handling load balancing. One is to let the PMC module software to apply a round robin schedule when transferring data to several CPUs, that is let the Video Handler software take on this task. My opinion is that this is ideal with a VME board, for VPX however it is possible to connect directly via the backplane, allowing the Video Multiplexer to adapt the round robin scheduler as for multi-board solutions. With the difference that Video Multiplexer here has to arrange data before addressing and transferring data direct to a single CPU on the board. Allowing the Video Handler to be removed and reducing the number of components.

Using A VPX board would however require more than just the board to be revised, either the rack has to be upgraded to support a hybrid CompactPCI/VPX backplane or all boards connecting to the CompactPCI backplane has to be changed or redesigned as well. Using VPX generates a lot more work to the operating environment but makes it possible to increase backplane bandwidth significantly.

10.3. GPP-GPU solution

Using a GPP-GPU solution is that alternative that will cause most work required for migration. The reasoning for this is that in moves away from the single SPE, single core model, instead a general purpose processor will do the non parallelised computations and then offload parallelised segments to the GPU.

Using GPU’s will also require the introduction of a new programming language to the software solution. Suitable programming language would be CUDA which has a strong standpoint in these types of applications. New DSP libraries and work related to this change will be required, also as for other alternatives a new IOP has to be implemented to run on one of the cores in the CPU.

For the considered GPP-GPU board there is no PMC/XMC sites available, so any communication with external boards will have to go via either 10 GE or x16 PCIe Gen 2 directly to the processor, thus the communicating boards has to be able to handle at least one of them.

11. Test Results

Practical testing was done in an isolated lab environment. Because of time constraints all hardware platforms where not tested. I choose to test CT12, a Intel Core i7-610E board from GE which hosted a VxWorks 6.8 environment and GE's DSP library, AXISLib-x86.

The new platform was chosen because I believe that test on a Core i7-610E would be the most rewarding.

Result where then compared to those on present platform, a summary of each test platform is given below.

Platform A:

- Present
- Board: CHAMP-AV
- CPU: MPC7410
- OS: VxWorks 5.4
- DSP Library: IXLibs-AV
- Backplane: VME

Platform B:

- New
- Board: CT12
- CPU: Single Core i7-610E board.
- OS: VxWorks 6.8
- DSP Library: AXISLib-x86
- Backplane: Compact PCI

11.1. IFFT Benchmark

The IFFT benchmarks where written in C with special SIMD [12] optimised methods performing IFFT's from different DSP libraries.

Elements	Platform A	Platform B	A/B
2^7	91.480 us	0.384 us	238.23
2^8	151.760 us	0.860 us	176.47
2^9	289.200 us	1.939 us	149.15
2^10	567.240 us	4.327 us	131.10
2^11	1155.120 us	10.508 us	109.93
2^12	2362.680 us	25.385 us	93.10
2^13	4718.080 us	64.742 us	72.88
2^14	9589.480 us	158.206 us	60.61
2^15	19360.039 us	403.438 us	47.99
2^16	41973.078 us	720.896 us	58.22
2^17	72926.641 us	1744.810 us	41.80
2^18	149474.766 us	3615.572 us	41.34
2^19	301088.813 us	9803.034 us	30.71

Table 11.1. *Execution times for IFFT's of different input sizes*

Results show a very good improvement for the new platform, by extracting the factor of increase it is easy to see that the improvement decrease for bigger inputs. Gaps become smaller as well and I believe that the factor of increase will be stabilised of even greater inputs. These are however not relevant and the IFFT benchmark cannot handle sizes bigger than 2^19.

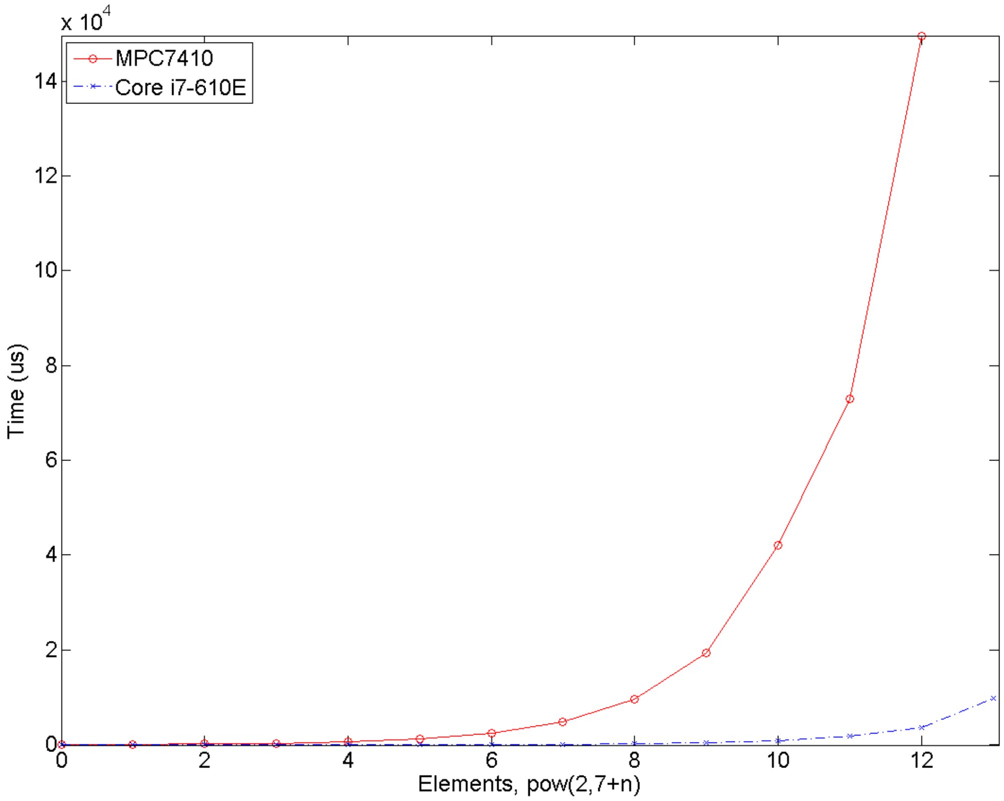


Diagram 11.1. *IFFT execution time for PowerPC and Intel.*

11.2. SVD Benchmark

The SVD benchmark was written in regular C code, no SIMD. This benchmark will give a good understanding of the increase for sequential applications.

Elements	Platform A	Platform B	A/B
1*(180*(24*18))	509 072 us	123 250 us	4.13
2*(180*(24*18))	1 015 639 us	245 827 us	4.13
3*(180*(24*18))	1 525 521 us	368 670 us	4.14
4*(180*(24*18))	2 028 773 us	491 519 us	4.13

Table 11.2. *SVD Execution times for 1-4 HRFC windows*

Results show more stable increase of performance compared to the IFFT benchmark. Increase can easily be derived to the increase clock frequency which is roughly five times higher for platform B, results show shows four times higher basic throughput.

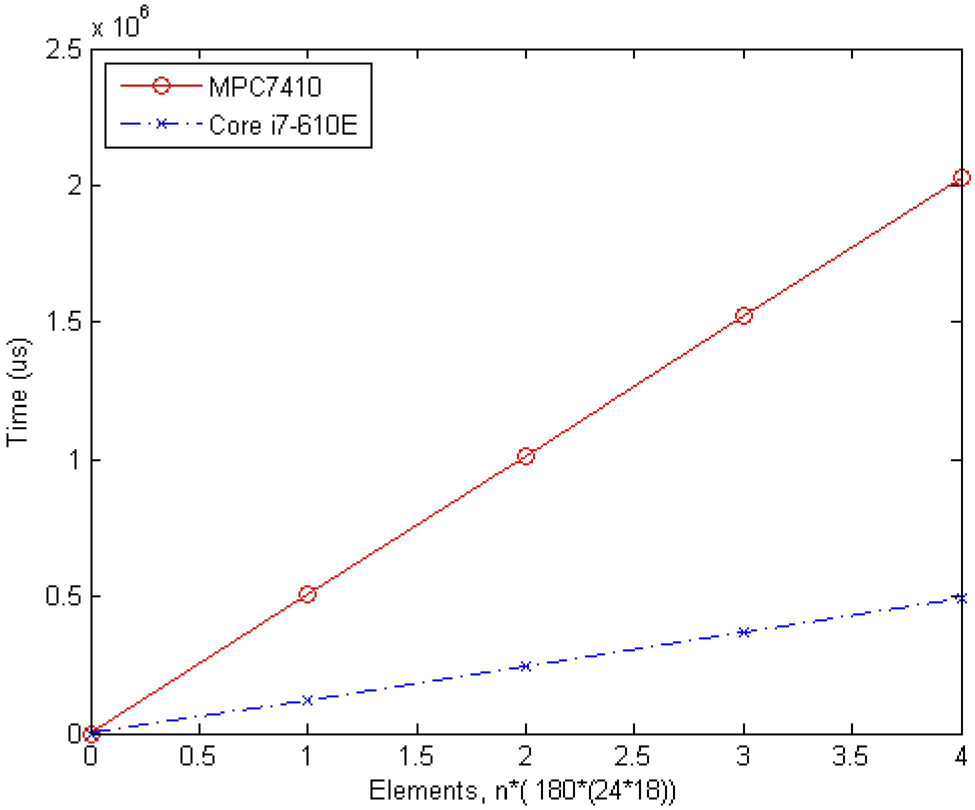


Figure 11.2. *IFFT execution time for PowerPC and Intel.*

12. RTOS comparsion

The time required for practical evaluation of different RTOS's did simply not exist so earlier test from internal and external sources has been used to give a brief measure of real-time performance.

- Montavista tests on a Dual Pentium M @ 2.0 GHz, showed 99.999% of the interrupt latencies where within 14 usec with worst-case latency 32 usec. [35]
- A comparison of Linux, VxWorks, Xenomai and RTAI on a MPC7445 @ 1 GHz shows that they all perform within the same regions. VxWorks had slightly lower latencies with only a few usec better. [34]

A performance evaluation was made on a SBC for a real-time application. The Linux variant used Linux kernel with PREEMPT_RT enabled, thus not suitable for hard real-time applications. However it is interesting to se how Linux compares to different hard RTOS's. Linux performs worse as load gets higher while the other OS's remains stable.

Interrupt latency represents the time from that an interrupt is received until the system responds, jitter is the additional time which measured times vary within.

System	**Linux**	**RTAI**	**Xenomai**	**VxWorks**
Jitter	0.40 us	0.15 us	0.25 us	0.50 us
Interrupt Latency	72.80 us	71.80 us	73.20 us	69.20 us

Table 12.1. *Interrupt latency.*

Real-time communication performance was evaluated by measureing the time between the input signal at one board and the output signal on another board connected with Ethernet. [34]

System	**Linux**	**RTAI**	**Xenomai**	**VxWorks**
Jitter	11.1 us	3 us	3.3 us	20.4 us
Latency	113 us	101 us	104.5 us	156.6 us

Table 12.2. *Real-time network communication latency.*

Tables show that for all for OS's values are within the same region except for network communication. Assuming that the relations from ***Table 12.1 and Table 12.2*** are somewhat accurate on an Intel processor as well, then there exist several alternatives to VxWorks.

With this knowledge the choice of RTOS becomes more a question about strategy and how to manage already acquired knowledge within current OS's.

Since these values are taken from two different evaluations and for two different architectures readers should only use this comparison as an in-formal guideline.

For further information about test look at references [34], [35]

13. Conclusion

There are several good alternatives to choose between.

- *PPC,* the alternative closest to what is used today. Allowing software to be reused without any significant changes and with a small increase in computational capacity. Has no implication on its environment, VME board with PCI interface. If time to market is more important than lifetime this would be optimal.
- *Intel,* the middle way alternative. A Intel platform will require mainly software re-design and implementation. There are two sources for this, new DSP libraries and adaption to the SMP model. Software updates required for one of the Intel alternatives is the same as for others with SMP, SMP will make it easy to move a application to a new general purpose CPU platform with none or little changes required. DSP libraries use the same APIs as long as hardware from the same vendor is used. All Intel SBC's except CHAMP-AV8 purchased for VME and VPX, choosing a VME SBC will not become a problem if a upgrade to VPX is done in the future. These properties make Intel platforms easy to maintain in the future.
- *GP-GPU,* the high performance and specialised alternative. To move to a GPU type of solution will require new programming languages, adaptation and verification of MUSIC on GPU and change of backplane amongst other changes. As for the Intel alternatives adaption to new DSP libraries has to be done. The GPU way differs a lot from the general purpose CPU alternative, thus a GPU solution would narrow down the number of future alternatives because the strongly coupled hardware architecture and software implementation. I also believe that the CPU is a possible bottleneck if floating point performance is considered the CPU/GPU ratio is 5/385. Some software segment will run better sequentially on a CPU, my concerns are that the CPU will not be able to handle the workload these imply for the GPU to be fully utilised.

Below is a grading of different parameters for each alternative, all grades are relative to the present platform. Because of the significant difference for A6 (GP-GPU), I have not found a proper way to grade it. Practical testing as only been done on one board, and test are for one core and the lowest measured performance increase.

SBC	**Peak FLOPS**	**Clk. Freq**	**Bwdth. Memory**	**Bwidth Backpl.**	**Power Cons.**	**SVD Test**	**IFFT Test**	**GFLOP/ Watt**
Pres.	1	1	1	1	1	1	1	1
A1	1.88	2.5	3.75	1.52	1.6	2*	2*	1.56
A2	1.25	5	32.25	7.58	1.3	4.13	30.7	1.94
A3	2.53	5	32.25	3.79	2	4.13	30.7	2.56
A3 (2 SBC's)	2.53	5	32.25	7.58	2.6	4.13	30.7	1.94
A4	5.85	4.2	39.75	7.58	1.5	4*	8*	5.59
A5	11.72	4.2	39.75	15.15	2.8	4*	8*	6.0
A5 (2 SBC's)	11.72	4.2	39.75	7.58	3.0	4*	8*	5.59
A6	24.07	-	-	15.15	1.8	-	-	13.84

***Estimated values**

Table 13.1. *Grading of alternatives (Values normalised to present CHAMP-AV solution).*

The reason for only testing one board is that boards for A4 and A5 just became available and time constraints.

13.1. Summary

This report concerns several different SBC’s, SIMD optimised libraries and RTOS’s. My opinion is that almost all of them are good depending on what outcome is desired. There is one with minimum impact, several different high performing alternatives and one which I believe is not so good because it requires more significant changes.

The minimal impact alternative uses processors with the same architecture as the present solution this makes software migration less demanding. It fits well into its working environment and is also scalable if extended functionality is desired. However this alternative uses relatively old processors, if a solution with longer lifetime is sought this may not be the optimal alternative. This is the PPC solution.

If higher performance solution is desired one way to go is away from PPC towards Intel. There are several different ways to go with Intel based SBC’s, those with modest performance and those which are high-performing. On benefit with moving to Intel is maintainability, it will be relatively easy to move software between different variations of Intel Core processors with varying number of cores etc. While keeping to the same vendor DSP libraries will come with the same API for both SSE 4.2 and AVX, making it easy to move software to new “Sandy bridge” processors and also utilise AVX performance (underlying libraries are changed, API remains the same). What Intel architecture to use, varies on demands, a more modest version can be applied to not only SP but also RC without over dimensioning. If a high performance is most important then a “Dual sandy” bridge solution would be suitable, requiring a overhaul of backplane and environment. With Intel the most possible variations of requirements can be satisfied and still not narrow down possibilities for future development.

The last way to go is with GP-GPU’s which has high theoretical performance, my opinion is that there are several drawbacks. First, all algorithms and software might not be optimal to run on a GPU, only those that are highly parallel. Second this platform would have the biggest impact not only to hardware and software implementation but most likely also for the underlying algorithms. It also takes a step away from the single thread model used for HR signal processing used today which not only is simple but also scales well to extended number of cores.

New general purpose processors enable computational capacity way beyond that of present solution, with no conceptual change to HRFC implementation. With GPU’s significant changes will have to be done, to hardware and software. Software changes required for GPU processing will narrow down future possible alternatives and with a doubtful outcome.

There exists several different Linux RTOS alternatives to VxWorks, which all show good real-time performance, competitive to VxWorks. Transition to new RTOS is rather a strategic decision than anything else with competitors as RTAI and Xenomai providing good real-time performance.

13.2. Proposal

I believe that the XVR14 is a good replacement of the current CHAMP-AV processor board. It is a single Intel Core i7-610E processor board. If a CPCI or VPX board is wanted then the *XCR14* or *VPXcel6 SBC624* SBC's from GE-IP could be considered, both with the same CPU as XVR14.

Parameters	**XVR14**
Computational capacity:	134.5 GFLOPS
CPU:	Core i7-2715QE
Clk. Frequency:	2.1 GHz
Cores:	4
SIMD instructions:	SSE 4.2, AVX
Memory: 16 GB DDR3	16 GB DDR3
Flash: 8 GB NAND	8 GB NAND
Mezzanines:	2 x PMC / XMC *(PCI-X / x8 PCIe Gen 2)*
Backplane	VME
Board Support Packages:	VxWorks, Linux, Windows
DSP Libraries:	AXISLib-x86

Table 13.2. *Summary of suitable board. [36]*

13.2.1. XVR 14

This is a high performance board which I think will replace the present HR signal processing board with good results. Work required for moving this processor board compared to VR12 will be none but with a more than three times the floating point performance compared. If less GFLOP's and higher clock frequency is desired the CPU could be exchanged for a Core i5 dual core at 2.5 GHz instead of 2.1 GHz. There are also more CPU alternatives.

The Core i7-2715QE has much higher floating point capacity than a dual Core i7-610E, my opinion is that it is better to use one single CPU board instead a dual CPU board or maybe two boards. However the XVR14 may not be as good if intensions are to use the same board elsewhere. The reason for this is that it will be very easy to over dimension. But since it is possible to choose between different "Sandy bridge" processors for the XVR12 I do not believe this should be a problem.

XVR14 is also a VME-board but GE has CPCI and VPX equivalents, XCR14 and VPXcel6 SBC624, thus implications for eventual upgrades will be minimal. At the moment GE has no dual CPU version of this board, hopefully they will, meaning that their products could cover all Intel variations considered. To be able to chose between several boards that all can run same software will make it very easy to build customised products with a high level of software reuse.

Predictions are that the number of cores will increase even more in future CPU's, with maybe hundreds of cores. Choosing a SMP OS will make software portable to these future developments by allowing more cores to be added and utilised without any software implications. Since this specific application benefits greatly of more independent and parallel tasks, this is something I believe should be prepared for.

13.2.2. Why not a Curtiss-Wright alternative

Of the considered processor boards GE-IP’s single boards provide higher PCIe bandwidth, 4 GB/s compared to 2 GB/s, which is the reason for not choosing the Curtiss-Wright 1905 board. Additionally GE has XVR14 in CPCI and VPX versions making future development and upgrades easier since DSP libraries can be left unchanged. There is currently no single “Sandy bridge” SBC from Curtiss-Wright either, I believe it is better to upgrade to a dual CPU board if needed rather than downgrading to a single CPU SBC.

14. Results and Discussion

I have chosen to mention quite a few variations of on hardware solutions, conceptually there are not so many, PPC, Intel and GPU. What is the optimal solution depends on several variables such as time-to-market, life-time, required performance, future plans and more. These are aspects which I, as a thesis student might not be know of.

One example of this is the choice of OS, there are many alternative and there are both pros and cons for all of them but without knowing the reasons for migrating to a new OS it is difficult to motivate why one is better than the other.

The two Intel solutions that are mentioned in the Proposals section are those that I believe can be implemented with good results. They have much in common and can easily be maintained and upgraded to multi-board or multi-CPU boards with the same processor architecture. If guidelines regarding software are followed, SMP and standardised DSP API's, then software will be able to move between the different variations of Intel based boards with very little effort. This makes them extremely generic since processor boards can be chosen specific for each product and software can be transferred between products with no big changes. It will also be possible to start with a VME board which later on can be upgraded to CPCI or VPX if desired with very small implications for signal processing software, required software upgrades will then only apply to I/O handling.

References

[1]. Manual for IXA7 Quad PowerPC Board Ver 2.4
(- - 1_1301-ROF61227 - en - A) (ORIGINAL).pdf, (Saab database)

[2]. Suppliers Datasheet for cPCI - 6190
(- - 1_1301-2_ROF61229 - en - A) (ORIGINAL).pdf, (Saab database)

[3]. Manual for PMC-GPIO Module Ver 2.0 (- - 1_1301-ROF1978504 - en - A) (ORIGINAL).pdf, (Saab database)

[4]. http://www.windriver.com / ; (2011-03-15)

[5]. http://www.mvista.com/real_time_linux.php ; (2011-03-15)

[6]. *www.windriver.com ; Wind River Linux and VxWorks Real-time capabilities: A Comparsion;* (2011-03-15)

[7]. http://www.windriver.com/products/product-overviews/realtime-core-for-linux_overview.pdf; (2011-03-15)

[8]. http://www.cs.ru.nl/lab/xenomai/exercises/ex01/Exercise-1.html; (2011-03-15)

[9]. http://www.dei.unipd.it/corsi/STR/RTAI/RTAI.pdf; (2011-03-15)

[10]. http://en.wikipedia.org/wiki/Symmetric_multiprocessing; (2011-03-15)

[11]. http://en.wikipedia.org/wiki/Asymmetric_multiprocessing; (2011-03-15)

[12]. http://defense.ge-ip.com/products/3535; (2011-03-15)

[13]. http://www.cwcembedded.com/continuum_vector.htm; (2011-03-15)

[14] http://www.vsipl.org/; (2011-03-15)

[15]. CAU1193403, Range bin selector. (Saab internal document)

[16]. Zito, Richard R.;A MUSIC (multiple signal classification) algorithm for specifying azimuth, elevation, and range of multiple sources
Acquisition, Tracking, and Pointing XX. Edited by Masten, Michael K.; Stockum, Larry A.; Thompson, William E.. Proceedings of the SPIE, Volume 6238, pp. 62380G (2006).

[17]. CAU1193404 MUSIC. (Saab internal document)

[18]. CAU1193405 High Resolution Extractor. (Saab internal document)

[19]. http://en.wikipedia.org/wiki/SISD; (2011-03-15)

[20]. http://en.wikipedia.org/wiki/SIMD; (2011-03-15)

[21]. http://en.wikipedia.org/wiki/MIMD; (2011-03-15)

[22]. http://en.wikipedia.org/wiki/Amdahl's_law; (2011-03-15)

[23]. http://en.wikipedia.org/wiki/Singular_value_decomposition; (2011-03-15)

[24]. http://www.ll.mit.edu/HPECchallenge/; (2011-03-15)

[25]. http://en.wikipedia.org/wiki/Fast_Fourier_transform; (2011-03-15)

[26]. http://en.wikipedia.org/wiki/Multiply-accumulate; (2011-03-15)

[27]. http://www.freescale.com/webapp/sps/site/overview.jsp?code=DRPPCALTVC; (2011-03-15)

[28]. White Paper: Intel x86-Based Digital Singal processing

[29]. http://software.intel.com/en-us/avx/; (2011-03-15)

[30]. CNU 113 146, Matlab tests of HRFC. (Saab internal document)

[31]. CAU 119 3404, MUSIC specification. (Saab internal document)

[32]. Eriksson, Per; Introduktion i radarsignalbehandling och radar styrning. (Saab internal document)

[33]. Barbalace, A.; Luchetta, A.; Manduchi, G.; Moro, M.; Soppelsa, A.; Taliercio, C.; , "Performance Comparison of VxWorks, Linux, RTAI, and Xenomai in a Hard Real-Time Application," Nuclear Science, IEEE Transactions on , vol.55, no.1, pp.435-439, Feb. 2008

[34]. CNU113145, DESIGN SPECIFICATION FOR FUNCTION BLOCK FIRE CONTROL EXTRACTOR IN GIRAFFE MK VI (Saab internal)

[35]. Weinberg, William; “Real-Time Technology for Embedded Linux: The MontaVista Advantage.

[36]. GE-IP; http://defense.ge-ip.com/ ; (2011-03-15)

[37]. Curtiss-Wright; http://www.cwcembedded.com/ ; (2011-03-15)

Appendix A – Suggested Boards

Boards considered for different variations of solutions.

Curtiss-Wright: CHAMP-AV4
Computational capacity: 40 GFLOPS
CPU: 4 x MPC7448 @ 1.25 GHz with Altivec *(Single Core)*
Memory: 512 MB DDR / CPU
Flash: 256 MB
Mezzanines: 2 x PMC / PMC-X *(PCI / PCI-X at 528 MB/s and 800 MB/s respectively)*
Backplane: VME
Board Support Packages: VxWorks, Linux
DSP Libraries: Continium Vector DSP Funtion Library

GE-IP: VR12
Computational capacity: 40.5 GFLOPS
CPU: 1 x Core i7-610E with SSE 4.2 *(Dual Core)*
Memory: 8GB DDR3
Flash: 16 GB NAND
Mezzanines: 2 x PMC-X / XMC *(PCI-X / x8 PCIe Gen2 at 1 GB/s and 4 GB/s respectively)*
Backplane: VME
Board Support Linux , Packages: Windows,VxWorks
DSP Libraries: AXISLib-x86

Curtiss-Wright: 1905
Computational capacity: 40.5 GFLOPS
CPU: 1 x Core i7-610E with SSE 4.2 *(Dual Core)*
Memory: 8 GB DDR3
Flash: 8 GB NAND
Mezzanines: 2 x PMC-X / XMC *(PCI-X / x8 PCIe Gen1 at 1 GB/s and 2 GB/s respectively)*
Backplane: VME, optional x4 PCIe to P0 connector
Board Support Packages: VxWorks, Linux,Windows
DSP Libraries: Continium Vector DSP Funtion Library

Curtiss-Wright: CHAMP-AV5
Computational capacity: 81 GFLOPS
CPU: 2 x Core i7-610E with SSE 4.2 *(Dual Core)*
Memory: 2 GB DDR3 / CPU
Flash: 8 GB NAND
Mezzanines: 1 x PMC/XMC + 1 x XMC *(PCI-X / x8 PCIe Gen1 at 800 MB/s and 2 GB/s respectively)*
Backplane: VME
Board Support Packages: VxWorks, Linux
DSP Libraries: Coninium Vexter DSP Funtion Library

<u>GE-IP: XVR14</u>

Computational capacity: 134.5 GFLOPS
CPU: Core i7-2715QE @ 2.1 GHz with SSE 4.2 and AVX *(Quad Core)*
Memory: 16 GB DDR3
Flash: 8 GB NAND
Mezzanines: 2 x PMC / XMC *(PCI-X / x8 PCIe Gen 2 at 1024 MB/s and 4 GB/s respectively*
Backplane: VME
Board Support Packages: VxWorks, Linux, Windows
DSP Libraries: AXISLib-x86

<u>Curtiss-Wright: CHAMP-AV8</u>

Computational capacity: 269 GFLOPS
CPU: 2 x Core i7-2715QE @ 2.1 GHz with SSE 4.2 and AVX *(Quad Core)*
Memory: 8 GB DDR3 / CPU
Flash: 8 GB NAND / CPU
Mezzanines: XMC *(x8 PCIe Gen2 at 4 GB/s)*
Data Plane / Expansion Plane interface: Gen2 sRIO / Gen 2 PCIe *(8 GB/s biderectional bandwidth)*
Backplane: VPX
Board Support Packages: VxWorks, Linux
DSP Libraries: Coninium Vexter DSP Funtion Library

<u>GE-IP: IPN250</u>

Computational capacity: 390 GFLOPS
CPU: SP9300 Core 2 Duo Penryn @ 2.26 GHz with SSE 4.1
GPU: NVIDIA GT240 96 core
Memory: 4 GB DDR3
Flash: 8 GB NAND
Mezzanines: None
Backplane: VPX
Expansion plane: x16 PCIe Gen 2 at 8 GB/s
Board Support Packages: VxWorks, Linux, Windows
DSP Libraries: AXISLib-x86

*PCI/PCIe rates are in each direction.

Appendix B – SBC overview

Summary of SBC's and properties considered to be of interest which has been investigated during this thesis.

Support		1905	C-AV4	C-AV5	C-AV8	VR12	XVR14	IPN250
Intel		X		X	X	X	X	X
Freescle			X					
Dual CPU's				X	X			
Single Core			X					X
Dual Core		X		X		X		X
Quad Core					X		X	
GPU/IGP					X		X	X
Altivec			X					
SSE 4.2		X		X		X		
AVX					X		X	
VME		X	X	X		X	X	
VPX					X			X
PMC		X	X	X		X	X	
PMC-X			X					
XMC		X		X	X	X	X	
PCI		X	X	X		X	X	
PCI-X		X	X	X		X	X	
PCIe	8xGen1						X	
	4xGen2			X				
	8xGen2	X				X	X	
	16xGen2				X			X
SerialRIO					X			
Gigabit Ethernet		X	X	X	X	X	X	X
Linux BSP		X	X	X	X	X	X	X
Windows BSP		X				X	X	X
VxWorks BSP		X	X	X	X	X	X	
IXLibs-AV, DSP		X						
AXISLib, DSP						X	X	X
Continium, DSP			X	X	X			

Table A1. *Characteristics of investigated SBC's.*

Appendix C – SBC performance overview

Support	**C-AV**	**C-AV4**	**C-AV5**	**C-AV8**	**VR12**	**XVR14**	**IPN250 ***
GFLOPS	16 GFLOPS	40 GFLOPS	81 GFLOPS	269 GFLOPS	40.5 GFLOPS	134.5 GFLOPS	390 GFLOPS
GFLOPS/W	0.32 GFLOPS/W	0.5 GFLOPS/W	0.82 GFLOPS/W	1.92 GFLOPS/W	0.62 GFLOPS/W	1.79 GFLOPS/W	4.33 GFLOPS/W
CPU's	4	4	2	2	1	1	1
Cores/CPU	1	1	2	4	2	4	2
Clk. Freq.	500 MHz	1.25 GHz	2.53 GHz	2.1 GHz	2.53 GHz	2.1 GHz	2.26 GHz
Memory	512 MB	512 MB	4 GB	16 GB	8 GB	16 GB	4 GB
Flash	16 MB	256 MB	8 GB	16 GB	16 GB	8 GB	8 GB
Max.backplane BW	528 MB/s	800 MB/s	2 GB/s	8 GB/s	4 GB/s	4 GB/s	8 GB/s
Max. Mem BW	528 MB/s	2 GB/s	17 GB/s	21 GB/s	17.1 GB/s	21 GB/s	-
TPD	50W	80W	100W	140W	65W	75W	90W

* GPU included.

Table A2. *Board performance summary.*

SBC	Peak FLOPS	Clk. Freq	Bwdth. Memory	Bwidth Backpl.	Power Cons.	SVD Test	IFFT Test	GFLOP/ Watt
C-AV	1	1	1	1	1	1	1	1
C-AV4	1.88	2.5	3.75	1.52	1.6	2*	2*	1.56
VR12	1.25	5	32.25	7.58	1.3	4.13	30.7	1.94
C-AV5	2.53	5	32.25	3.79	2	4.13	30.7	2.56
XVR14	5.85	4.2	39.75	7.58	1.5	4*	8*	5.59
C-AV8	11.72	4.2	39.75	15.15	2.8	4*	8*	6.0
IPN250	24.07	-	-	15.15	1.8	-	-	13.84

*Values are theoretical estimations

Table A3. *Grading relative to present CHAMP-A (Normalised).*

Appendix D – Used Altivec optimised functions

Summary of vector optimised functions that have been used for implementation of MUSIC SWU.

Function	Description
mcomplex	Create new complex matrix
mcopyz	Fill matrix with scalar (= 0 for a reset)
vfftinit	Init the fft, with argument = length of IFFT
mconj	Conjugate of a matrix. (Used on Es)
mmultcols	Matrix column / Vector element-wise multiplication (Multiply steering vector g with mconj(Es))
mifftcols	IFFT of each column of a matrix (Used on mmultcols(Es,g))
mmag2	calculate squared magnitude of each element in a matrix(Used on mifftcols(f))
mgetrow	Get a row of a matrix (used after mmag2(P))
velsum	Sum all elements of a vector (used on each row on mmag2(P))

Table A3. *Used functions from IXLibs-av for MUSIC SWU*

Appendix E – Formula for peak performance

A: Number of cores
B: Number of execution units
C: Cycles per second
D: Execution unit operations / cycle
E: Floats per execution

Result : A*B*C*D*E

Example MPC7410:
A = 1
B = 2
C = 500 MHz
D = 1
E = 4

(1,2,500E6,1,4) = 4 GFLOPS