AVSynDEx Methodology For Fast Prototyping of Multi-C6x DSP Architectures

Jean-François NEZAN, Virginie FRESSE, Olivier DEFORGES, Michael RAULET
CNRS UMR IETR *(Institut en Electronique et Télecommunications de Rennes)*
INSA RENNES
20, av des Buttes de Coësmes, CS 14315
35043 RENNES Cedex, France
jnezan@insa-rennes.fr
Phone number : +33/0 233 238 459 Fax : +33/0 223 238 262

Keywords : parallel architectures, real time applications, fast prototyping, LAR image coding.

Abstract

In this paper we present AVSynDEx (concatenation of AVS+SynDEx), a rapid prototyping process for the implementation of digital signal processing applications on parallel architectures. This process is based on the use of widely available and efficient CAD tools established along the design process so that most of the implementation tasks become automatic. One main advantage is that only a signal-processing designer is needed, all the other specialized manual tasks being transparent in this prototyping methodology, hereby reducing the implementation time.

We have developed executive kernel for the C6X DSP family, in order to automatically generate a distributed and optimized static executive of the specified algorithm for new generation processors. We demonstrate the efficiency of our methodology with the development of a LAR image coding application and its multi-C6x DSP implementation.

1 Introduction

The prolific evolution of telecommunication, wireless and multimedia technologies has sustained the requirement for the development of increasingly complex integrated systems. Indeed, digital signal processing applications, including image processing have become more and more complex, thereby demanding much greater computational performances. This aspect is especially crucial for certain real-time applications: to validate a new technique only functionality is not sufficient, the algorithm has to be executed in a limited time. Until then the first approach to meet this aspect was to optimize the algorithm, a digital signal or image designer could do this task. Nevertheless, this solution was quickly inadequate and parallel to the algorithm development, the implementation aspect must be taken into account.

The objective of this work is to propose a full rapid prototyping process (AVSynDEx) by means of both existing academic and commercial CAD tools and platforms. A translator between the CAD environments provides automatic stages in the implementation process.

The prototyping methodology enables a single digital signal or image-processing designer to create the application with a traditional development environment (AVS). Then he supervises the implementation on a parallel architecture without any other necessary skills. This methodology ensures an easy handing-over up to date of the algorithms. AVSynDEx is open-ended : other CAD environments or architectures can be integrated. Another advantage is the partitioning decision between target DSPs at the highest level of the application description. Then the integration of a distributed executive generator (SynDEx) leads to an optimized implementation on a parallel and mixed platform. Furthermore, the same programmable platform can be used for several applications.

SynDEx can handle different processors: Analog Device ADSP 21060, SHARC, Motorola MPC 555 and MC 68332, Intel i80x86 et i8096, Unix/Linux workstations, and Texas Instruments TMS320C40 (the latter of which is no longer efficient enough for new applications). Our aim is to combine the advantages of SynDEx with C6x DSP performances, creating a SynDEx automatic code generator.

This paper is organized into 4 sections. In Section 2, the AVSynDEx methodology is described and the executive kernel for C6x DSPs is explained in section 3. Results of a LAR coding process implementation are developed in section 4. Finally, conclusions and perspectives are given in section 5.

2 AVSynDEx methodology

The methodology dedicated to homogeneous parallel DSP architectures is first described [], and the two efficient CAD tools used: AVS and SynDEx.

2.1 AVS: Advanced Visual System

AVS [3] is a multi-platform, component-based software environment building applications with interactive visualization and graphics features. AVS employs an innovative object-oriented visual programming interface to create, modify, and connect application components.

The top section of the AVS Network Editor, shown Fig 1, consists of libraries of objects. The bottom section is the workspace where the user drags and drops objects and draws connection lines between objects to assemble an application or to create new objects. Objects are stored in new libraries for future applications. AVS visualization tools and the error dialog box are used to observe calculations. Objects created are linked to C source code developed, compiled and debugged using Microsoft Visual C++ tools.

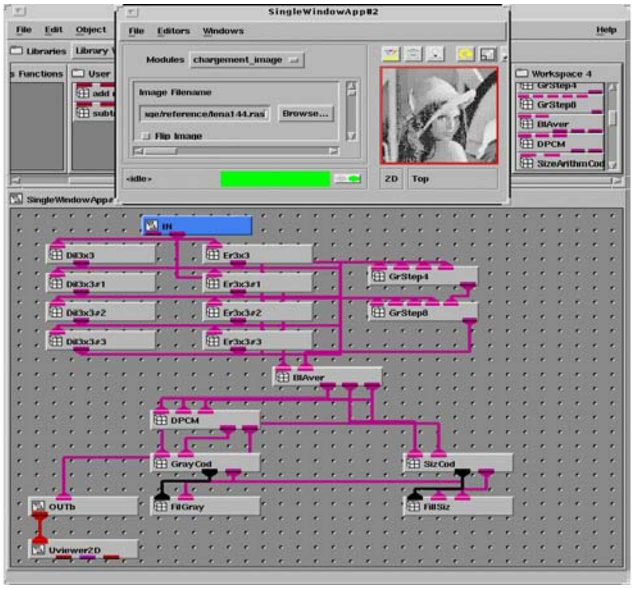


Fig 1 : LAR algorithm under AVS environment

An automatic translator is designed for the generation of a global SynDEx description from the data flow graph created with AVS, with "cleaning" performed on the AVS source functions to provide standard C code.

2.2 SynDEx

SynDEx [4] (SYNchronized Distributed Executive) is a free academic CAD software system developed by INRIA Rocquencourt, France. It supports the AAA methodology (Adequation Algorithme Architecture). Adequation is a french word meaning efficient matching. The goal of Adequation is to find the best matching between an algorithm and an architecture. The AAA optimization heuristic handles both heterogeneous architectures and inter-processor communications.

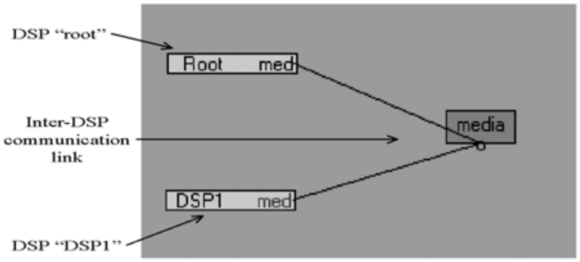


Fig 2 : SynDEx architecture graph

On the one hand, SynDEx uses a material graph, which models the multiprocessor architecture. Fig 2 shows an architecture made of two DSP ("root" and "DSP1") connected each other with one media (called "media"). On the other hand, a software graph describes the dataflow graph (Fig 3).

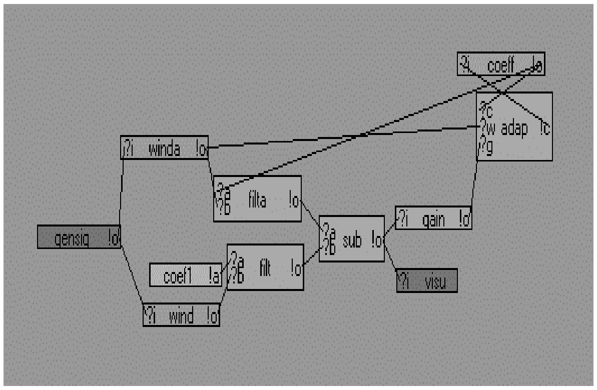


Fig 3 : SynDEx algorithm graph

SynDEx then carries out the placement and partitioning, according to the time spent for data transfers between processors, and for each task of the algorithm. The result can be visualized and analyzed thanks to the timing diagram generated by SynDEx, shown in Fig 4.

SynDEx generates an executive into several source files, one for each processor of the architecture, and another one for automating the architecture specific compilation chain. The code generated can insert chronometrical reports for heuristic optimization, for the final implementation. The main advantage of SynDEx is to avoid operating system like 3L diamond product, implementing the minimum custom-built static executive needed for a given application. Spatial and temporal additional costs are minimized, whereas the order of algorithm tasks is guaranteed and locking is avoided.

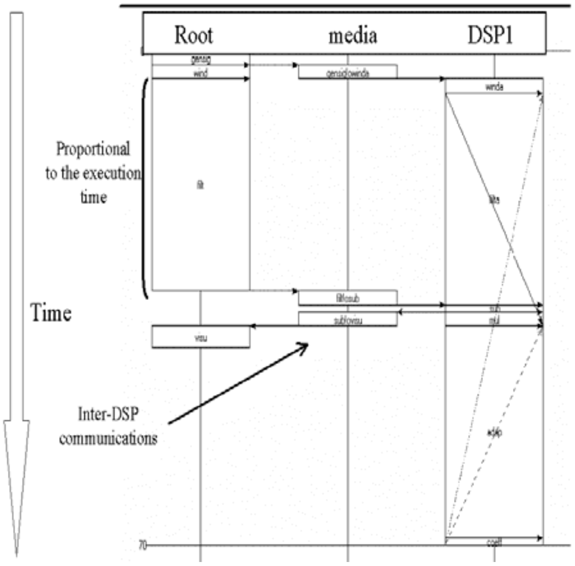


Fig 4 : SynDEx timing graph

2.3 AVSynDEx

The prototyping process starts with the functional description of the application, using AVS. This description is then automatically translated into an input file for Syndex, and is used initially to determine the time associated with each function. It creates a monoprocessor implementation and the user can achieve chronometrical reports using the target processor debugging tool (Code Composer for C6x). The user can easily copy out these times into the software SynDEx graph. Syndex then generates a real-time distributed and optimized executive specific to the target platform.

The main advantage of this prototyping process is its simplicity, as most of the tasks are performed within the user conventional environment. The required knowledge of SynDEx and the loader are limited to simple operations.

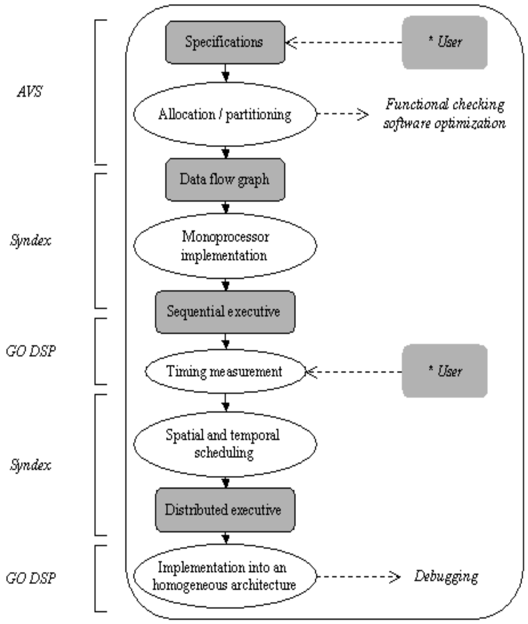


Fig 5 : rapid prototyping methodology for parallel architectures

Automatic generation code for Texas Instruments C6x DSPs have been developed because required computational performances should always be improved for a real time use.

3 SynDEx kernel for C6x DSP family code generation

3.1 TMS320C6x DSP properties

The C6X VLIW architecture operates between clock rates of 150 to 600 MHz, thereby supplying up to eight 32-bit instructions, to the eight functional units every clock cycle. C6x are high-performance DSPs, but are not dedicated to multiprocessor architectures. Therefore, manufacturers must insert additional digital resources between two C6x processors to enable their communication between (Fig 6). Inter-C6x communications are then architecture dependant.

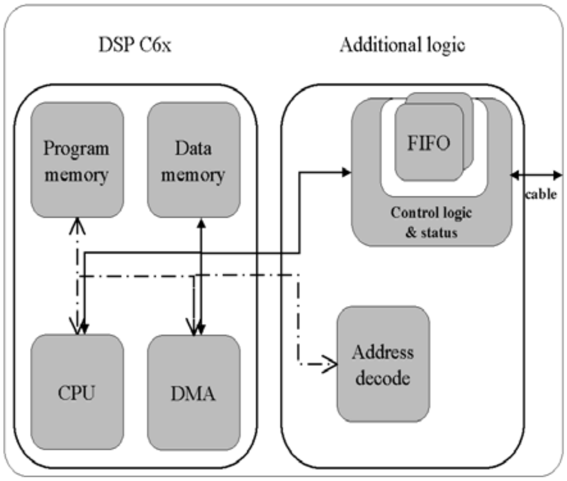


Fig 6 : C6x additional logic for communications

C40 DSPs contain communication ports (CPs), providing inter-C40 communication. This specific interface allows the simple construction of processor systems. Moreover, all of this was written in the C40 specific assembly language. As a result, SynDEx C40 executive kernel can no longer be used for C6x.

3.2 Developed kernel

The executive generated by SynDEx is divided into several source files, each containing intermediate code composed of a list of macro-calls of the intermediate generic kernel SynDEx. Those macro-calls will be translated by the macro processor M4 [5] into a source code in the compilable language for target processor. We have created M4 libraries for C6x forming the SynDEx C6x Kernel.

Code Composer Studio software [6] speeds up and improve the C6x development process for programmers. It accepts C source code, which is optimized to high performance without the programmer requiring assembly language knowledge that is specific and complex to the VLIW architecture. It was decided to develop C6x Kernel in C language, in order to make it partially reusable for any C programmable DSP.

The C6x executive kernel has been divided into several libraries (Fig 7), enabling its easy adaptation to a new architecture. The non-generic library contains M4 macros for the application, such as specific input/output functions. The architecture independent library contains macros used whatever the architecture target. The other libraries are architecture-dependent : processor type, processor or communication type dependent.

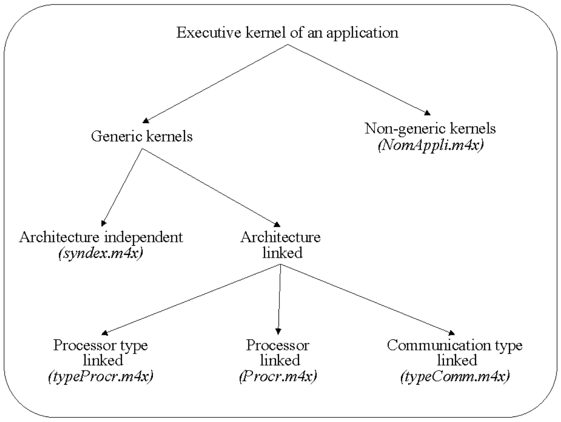


Fig 7 : Executive Kernel Organization

Kernels have been developed for two different platforms proving that macros can be reused for every platform. The adaptation of our work for another multi-C6x architecture is restricted to the communication sequence adaptation for a new media, if needed.

SynDEx macrocode creates two interleaved schedulers: one for computation tasks and the other for communications, allowing parallelism of those actions. The choice of the multi-channel DMA transfer use (Fig 8) maximizes the parallelism and timing performances.

The DMA contains four different channels, each communication media are linked to one channel. SynDEx architecture graph can have four connections for each C6x.

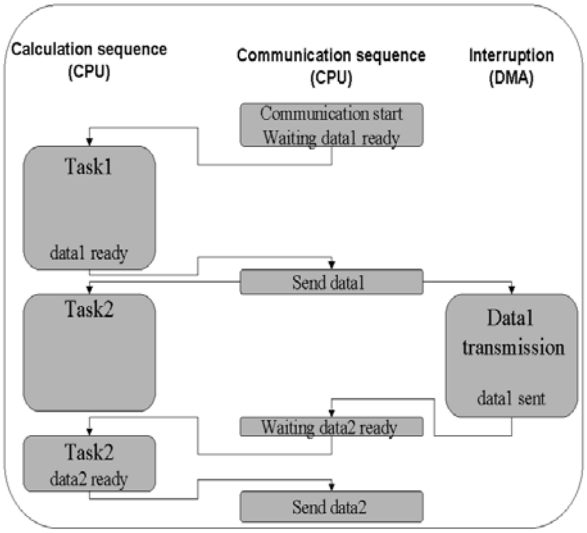


Fig 8 : Parallelism Computation / Communication

4 LAR application

A new image compression algorithm has been developed in our laboratory: its implementation on a multi-C6x architecture provides a validation of our fast prototyping methodology. This LAR algorithm, meaning Locally Adaptive Resolution [7], is an efficient technique well suited for image transmission via Internet or for embedded systems.

The basic idea of the LAR method is that the local resolution (pixel size) can depend on the activity: when the luminance is locally uniform, the resolution can be low (large pixel size). When the activity is high the resolution has to be finer (smaller pixel size). The image is first subsampled by 16x16 blocks representing local trees. Each one is then split according to a quadtree, dependent on the local activity, estimated from morphological gradient (dilatation and erosion) measures. The finest resolution is typically 2x2 blocks. The image can be reconstructed by associating the corresponding average luminance in the source image to each block.

The image contents information given through the block size is considered advantageous for the luminance quantization, during a predictor encoding scheme (DPCM). Large blocks require a fine quantization as they are located in uniform area (strong sensitivity of human eye to brightness variations). Small ones support a coarse quantization as they are upon edges (low sensitivity). Size and luminance are both encoded by an adaptive arithmetic entropic encoder. The average cost is less than 4 bits per square.

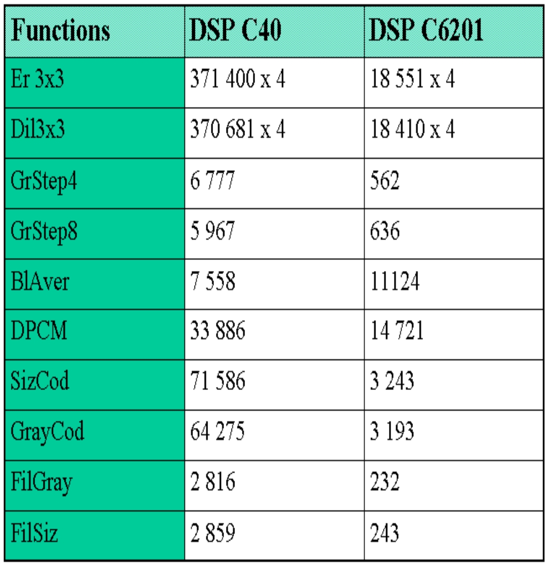
4.1 AVS modules and application development

The algorithm development and optimizations are achieved using the AVS tool. The designer can develop the application and can easily refine the granularity of several functions. The data flow graph algorithm of the LAR application is developed and the resulting AVS data flow graph is shown in Fig 1. AVS enables the image-processing designer to check the functionality of the new algorithm.

4.2 DSP implementations

Next, SynDEx is used for the generation of a monoprocessor implementation of the application. Fig 9 displays the chronometrical reports for a single C40 DSP (40 MHz) and with one C6201 DSP (200 MHz clock frequency) of our SUNDANCE SMT320 Motherboard. The global resulting time for the C40 is 3163ms, and 182ms for the C6201, the accelerating factor being 17.3. The VLIW architecture of the C6x leads to this significant acceleration.

These execution times are reported in SynDEx graph for the generation of the real-time distributed executive for our two C6201 DSPs platform. The resulting time is 116ms, and the accelerating factor that is 1.57.



Functions	DSP C40	DSP C6201
Er 3x3	371 400 x 4	18 551 x 4
Dil3x3	370 681 x 4	18 410 x 4
GrStep4	6 777	562
GrStep8	5 967	636
BlAver	7 558	11124
DPCM	33 886	14 721
SizCod	71 586	3 243
GrayCod	64 275	3 193
FilGray	2 816	232
FilSiz	2 859	243

Fig 9 : Chronometrical reports for the LAR application (microsec)

5 Conclusions and perspectives

This paper has demonstrated the efficiency of the AVSynDEx methodology for multiprocessor platforms. Modules developed can be modified and connected each other at a high level. Functionality of a new application can be checked with interactive visualization and graphic features provided by AVS. The user can then project its applications onto a complex architecture, without any implementation prerequirements.

An automatic distributed executive generator for multi-C6x DSP architectures using SynDEx is created. It was tested on Texas Instruments TMS320C6201, but it is suitable for other C6x family's DSPs. Furthermore, the use of C language can be a basis for other DSP kernel developments. Static executives generated are custom-built, avoiding adding operating systems and saving platform resources.

Implementing the LAR image coding application demonstrates that this prototyping process is fast, secure and can be handled by a single digital signal or image designer. An Mpeg-4 decoder and a software radio applications will soon take advantage of it [8].

We are working with SynDEx V6 developers on the automatic code generator integrating new features: shared memory, conditional nodes and hierarchy in the SynDEx algorithm graph description. The last issue will ensure the description and implementation of application with partial or full reconfigurability.

The proposed methodology can be extend to FPGA and DSP architectures, as shown in [2]. The additional logic added between two C6x DSPs is often integrated in a FPGA. The implementation of elementary and regular operations onto this material part would give higher performances. So, we plan to study the adding of this material element in the SynDEx material graph, and the generation code for FPGA.

References

[1] V. Fresse, M. Assouil, O. Deforges : *Rapid prototyping of image processing onto a multiprocessor architecture*. DSP World ICSPAT, Orlando, Florida, USA, Nov 99.

[2] V. Fresse, O. Deforges, M. Assouil *: Rapid prototyping for mixed architectures*, IEEE International Conference on Acoustics, Speech, and Signal Processing, Istanbul, Turkey, June 00.

[3] International AVS Center, Manchester Visualization Centre, Manchester Computing, University of Manchester. Available at http://www.iavsc.org.

[4] T.Grandpierre, C.Lavarenne, Y.Sorel : *optimized rapid prototyping for real time embedded heterogeneous multiprocessors*. In 7th International workshop on Hardware/Software Co-De*sign*. IEEE Computer Society, ACM SIGSOFT, IFIP. Rome, Italy, May 99.

[5] M.Loukides, A.ORAM. *Programming with GNU software*. O'Reilly, 1996.

[6] Texas Instruments technical documents : *TMS320C6000 Code Composer Studio – tutorial*. Ref spru301c. Available at http://www.dspvillage.ti.com. Feb 00.

[7] O.Deforges, L.Bedat, J.Ronsin : *Low Bit-Rate Compression Based on LAR Method for Video-Conference via Internet*. 1st IEEE International Conference on Multimedia and Expos (ICME), New York, USA, Aug 00.

[8] A.Kountouris, C.Moy, L.Rambaud, P.Lecorre : *Reconfigurable radio case study : A software based multi-standard transceiver for UMTS, GSM, EDGE and Bluetooth*. In IEEE VTC (Vehicular Technology Conference) Fall'2001, Atlantic City, USA, Oct 01.