3D-SoftChip: A Novel 3D Vertically Integrated Adaptive Computing System

Chul Kim[1], Alex Rassau[1], Stefan Lachowicz[1], Saeid Nooshabadi[2] and Kamran Eshraghian[3]

1 Centre for Very High Speed Microelectronic Systems, School of Engineering and Mathematics, Edith Cowan University, Perth, WA, 6027 Australia {c.kim, a.rassau, s.lachowicz}@ecu.edu.au,
WWW home page: http://www.soem.ecu.edu.au

2 School of Electrical Engineering and Telecommunications, The University of New South Wales, Sydney, NSW, 2052 Australia
saeid@unsw.edu.au
WWW home page: http://www.eet.unsw.edu.au

3 Eshraghian Laboratories Pty Ltd, Technology Park, Bentley, WA, 6102 Australia
k.eshraghian@elabs.com.au
WWW home page: http://www.elabs.com.au

Abstract. This paper describes the high-level system modeling and functional verification of a novel 3D vertically integrated Adaptive Computing System-on-Chip (ACSoC), which we term 3D-SoftChip. The 3D-SoftChip comprises two vertically integrated chips (a Configurable Array Processor and an Intelligent Configurable Switch) through an Indium Bump Interconnection Array (IBIA). This paper also describes an advanced HW/SW co-design and verification methodology using SystemC, which has been used to verify the functionality of the system and to allow architectural exploration in the early design stage. An implementation of the MPEG-4 full search block matching motion estimation algorithm has been applied to demonstrate the architectural superiority of the proposed novel 3D-ACSoC.

1 Introduction

As the microelectronics industry enters the nano and giga-scaled integrated circuit era, system design is becoming increasingly challenging as the complexity of integrated circuits (ICs) rises exponentially. The keenly shortened time-to-market period and relentlessly increased non-recurring engineering (NRE) cost are also becoming ever more problematic factors. Another growing problem is related to interconnection densities, as semiconductor geometries continue to shrink the system

performance of ICs is increasingly dominated by interconnection performance. Moreover, most current systems have highly demanding data bandwidth requirements, particularly for real-time communication or video processing applications. To address this interconnection and system-on-chip complexity crisis, innovative new computing systems with novel interconnection methods will be required. A very promising candidate to overcome these problems is the concept of a 3D integrated adaptive computing system-on-chip (3D-ACSoC). This concept may well be a critical technology for the next generation of computing systems because of its wide applicability/adaptability and because of the significant benefits gained from 3D systems such as a reduction in interconnect delays and densities, and reduction in chip areas due to the possibility for more efficient layouts etc. This paper describes the modeling and functional verification of such a 3D-ACSoC, the 3D-SoftChip [1, 2].

Conventional SoC design methodologies include many error-prone and tedious iteration processes, which can result in a lack of system reliability and extend the design time. Moreover, the portion taken up by verification processes in the total design time is exponentially increasing. By adopting the proposed SoC design methodology using SystemC, the design time can be significantly reduced and more reliable systems can be realised.

Figure 1 illustrates the physical architecture of the 3D-SoftChip comprising the vertical integration of two 2D chips. The upper chip is the Intelligent Configurable Switch (ICS). The lower chip is the Configurable Array Processor (CAP). Interconnection between the two planar chips is achieved via an array of indium bump interconnections.

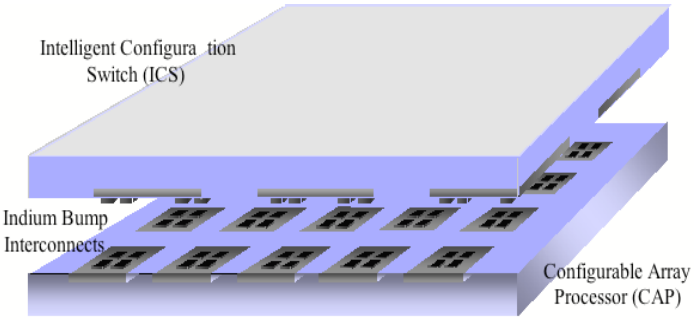


Fig. 1. . 3D-SoftChip Physical Architecture

The rest of the paper is organized as follows: Section 2 introduces an overview of 3D adaptive computing systems. Section 3 describes the overall architecture and the salient features of 3D-SoftChip. A suggested HW/SW co-design and verification methodology for development of the 3D-SoftChip is described in Section 4. Section 5 provides high-level modeling using SystemC and application mapping. Finally, some conclusions are made in Section 6.

2 3D Adaptive Computing Systems

2.1 3D-SoC Overview

3D systems are becoming an increasingly promising technology to combat the current wiring crisis. Previous work has shown that the 3D integration of systems has a number of benefits [3, 4]. As described by Joyner, et al, 3D system integration offers a 3.9 times increase in wire-limited clock frequency, an 84% decrease in wire-limited area or a 25% decrease in the number of metal levels required per stratum. There are three feasible 3D integration methods; a stacking of packages, a stacking of ICs and Vertical System Integration as was introduced by IMEC [5]. There are four main enabling technologies for the fabrication of 3D-ICs, Beam recrystallization, Silicon Epitaxial Growth, Solid Phase Crystallization and Processed Wafer Bonding [6].

Table 1. 3D Fabrication Technologies

3D Fabrication Technologies	Characteristics
Beam Recrystallization	Deposit poly-silicon and fabricate Thin-Film Transistors (TFTs) High Performance of TFT's The high melting temperature of poly-silicon means it is probably not a practical fabrication technology Suffers from low carrier mobility
Silicon Epitaxial (SE) Growth	Epitaxially grow a single crystal Si High temperature causes degradation in quality of devices Process not yet manufacturable
Solid Phase Crystallization	Low temperature alternative to SE Flexibility of creating multiple layers Compatible with current processing environments Useful for stacked SRAM and EEPROM cells
Processed Wafer Bonding	Bond two fully processed wafer together Similar electrical properties on all devices Independent of temperature since all chips are fabricated then bonded Good for applications where chips do independent processing Lack of precision (alignment) restricts inter-chip communication to global metal line

Table 1 shows the main characteristics of each of these 3D fabrication technologies. In this research, however, the focus is on an indium bump interconnection array (IBIA). The reason why wafer bonding technology is adopted for this work is because the process has particular benefits for applications where each chip carries out independent processing. The characteristic of the 3D-SoftChip is that each of the two planar chips should be effectively manipulated to maximize

computation throughput with parallelism. The use of 3D flip-chip wafer bonding technology allows relatively easy signal distribution because signal connections can be made between the two vertically integrated planar chips. Moreover, it has low parasitics (inductance, capacitance), and up to four orders of magnitudes smaller RC parameters, allowing fast signal transmission over a large chip area with little attenuation and minimum global clock skew while local clock skew is also kept low. Indium is chosen for the interconnects as it has good adhesion, a low contact resistance and can be readily utilized to achieve an interconnect array with a pitch as low as 10μm. The development of 3D integrated systems will allow improvements in packaging cost, performance, reliability and a reduction in the size of the chips.

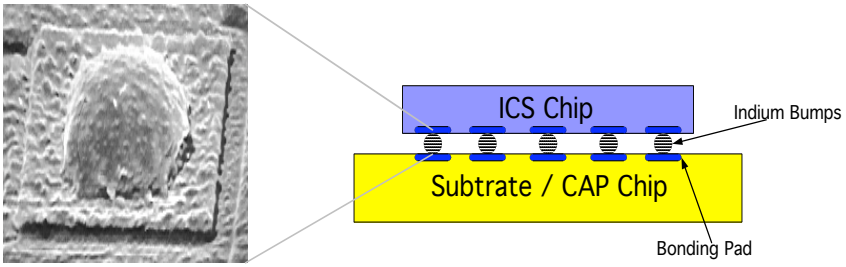


Fig. 2. . 3D Flip-Chip Wafer Bonding Technology using Indium Bump Interconnection Array (IBIA)

2.2 Adaptive Computing Systems

A reconfigurable system is one that has reconfigurable hardware resources that can be adapted to the application currently under execution, thus providing the possibility to customize across multiple standards and/or applications. In most of the previous research in this area, the concepts of reconfigurable and adaptive computing have been described interchangeably. In this paper, however, these two concepts will be more specifically described and differentiated. Adaptive computing will be treated as a more extended and advanced concept of reconfigurable computing. Adaptive computing will include more advanced software technology to effectively manipulate more advanced reconfigurable hardware resources in order to support fast and seamless execution across many applications. Table 2 shows the differentiations between reconfigurable computing and adaptive computing

Table 2. Reconfigurable Vs Adaptive Computing Systems

	Reconfigurable Systems	Adaptive Computing Systems
Hardware Resources	Linear array of homogeneous elements(Logic gates, look-up tables)	Heterogeneous algorithmic elements(Complete function units such as ALU, Multiplier)

	Reconfigurable Systems	Adaptive Computing Systems
Configuration	Static, Dynamic Configuration, Slow reconfiguration time	Dynamic, Partial run-time reconfiguration
Mapping Methods	Manual routing, conventional ASIC design tools (HDL)	High-level language (SystemC, C)
Characteristics	Large Silicon area, Low speed(High capacitance), High power consumption, High cost	Smaller Silicon size, High speed, High performance, Low power consumption, Low cost

2.3 Previous Work

Adaptive computing systems are mainly classified in terms of granularity, programmability, reconfigurability, computational methods and target applications. The nature of recent research work in this area according to these classifications is shown in Table 3.

Table 3. Reconfigurable and Adaptive Computing Systems

System	Granularity / PE Type	Programmability	Reconfiguration	Computation Method	Target Application
RapiD [7]	Coarse(16bits), Homogeneous	Single	Static	Linear Array	Systolic arrays, Data-intensive
RAW [8]	Mixed, Homogeneous	Single	Static	MIMD	General purpose
MorphoSys [9]	Coarse(16bits), Homogeneous	Multiple	Dynamic	SIMD	Data-parallel, Computation intensive app.
QuickSilver Adapt2400 [10]	Coarse(8,16,24,32bits), Heterogeneous	Multiple	Dynamic	Heterogeneous Node Array	Comm. Multimedia DSP
Elixent DFA1000 [11]	Coarse(4bits), Heterogeneous	Multiple	Dynamic	Linear D-Fabric Array	Multimedia app.
picoChip PC102 [12]	Coarse(16bits), Heterogeneous	Multiple	Dynamic	3way-LIW	Wireless Comm.
3D-SoftChip	***Coarse(4bits), Heterogeneous***	***Multiple***	***Dynamic***	***Various types of computation models***	***Comm. Multimedia DSP***

This table shows that the early research and development was into single linear array type reconfigurable systems with single and static configuration but that this

has evolved towards large adaptive SoCs with heterogeneous types of reconfigurable hardware resources and with multiple and dynamic configurability. As illustrated above, the 3D-SoftChip has several architectural superiorities when compared with conventional reconfigurable / adaptive computing systems resulting from the 3D vertical interconnections and the use of state of the art adaptive computing technology. This makes it highly suitable for the next generation of adaptive computing systems.

3 3D-SoftChip Architecture

Figure 3 shows the overall architecture of the 3D-SoftChip. As can be seen, it is comprised of 4 UnitChips. Each UnitChip has 16 sets of heterogeneous arrays of Processing Element (PE), a 32-bit dedicated RISC control processor and a high bandwidth data interface unit. A more detailed description of the architecture and interconnection network can be seen in [1].

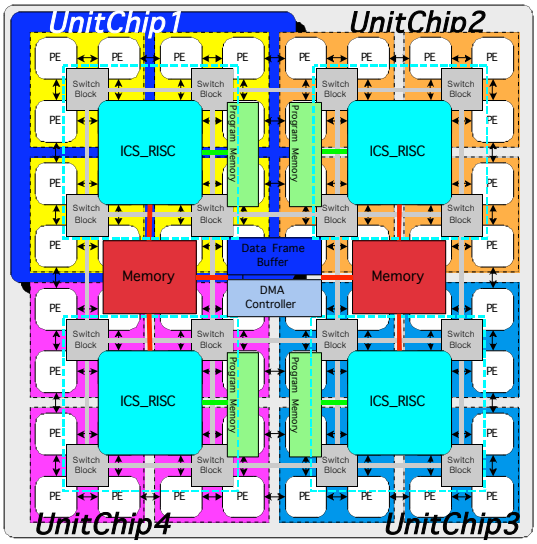


Fig. 3. .Overall Architecture for 3D-SoftChip

3.1 Overall Architecture of 3D-SoftChip

According to a given application program, the PE array processes large amounts of data in parallel while the ICS controls the overall system and directs the PE array execution and data and address transfers within the system.

3.2 Overall Architecture of 3D-SoftChip

The 3D-SoftChip has 4 distinctive features: Various types of computation model, adaptive word-length configuration [2, 13], optimized system architecture for communication and multimedia signal processing and dynamic reconfigurablility for adaptive computing.

3.2.1 Computation Algorithm

As described above, one 32-bit RISC controller can supply control, data and instruction addresses to 16 sets of PEs through the completely freely controllable switch block so various computation models can be achieved such as SISD, SIMD, MISD, MIMD as required. Enough flexibility is thus achieved for an adaptive computing system. In the SIMD computation model, 3 types of different SIMD computation can be realized; massively parallel, multithreaded and pipelined [14]. In the massively parallel SIMD computation model, each UnitChip operates with the same global program memory. Every computation is processed in parallel, maximizing computational throughput. In the Multithreaded SIMD computation model, the executed program instructions in each UnitChip can be different from the others, so multithreaded programs can be executed. The final one is the pipelined SIMD computation model. In this case each UnitChip executes a different pipelined stage

3.2.2 Word-length Configuration

This is a key characteristic in order to classify the 3D-SoftChip as an adaptive computing system. Each PE's basic processing word-length is 4-bit. This can, however, be configured up to 32-bit according to the application in the program memory. This flexibility is possible due to the configurable nature of the arithmetic primitives in the PEs [13] and the completely freely controllable switch block architecture in the ICS chip.

3.2.3 Optimized System Architecture for Communication and Multimedia Signal Processing

There are many similarities between communications and multimedia signal processing, such as data parallelism, low precision data and high computation rates. The different characteristics of communication signal processing are basically more data reorganization, such as matrix transposition, and potentially higher bit level computation. To fulfill these signal processing demands, each UnitChip contains two types of PE. One is a standard-PE for generic ALU functions, which is optimized for bit-level computation, the other is a processing accelerator-PE for DSP. In addition, special addressing modes to leverage the localized memory along with 16 sets of loop buffers to generate iterative addresses in the ICS add to the specialized characteristics for optimized communication and multimedia signal processing

3.2.4 Dynamic Reconfigurability for Adaptive Computing

Every PE contains a small quantity of local embedded SRAM memory and additionally the ICS chip has an abundant memory capacity directly addressable

from the PEs via the IBIA. Multiple sets of program memory, the abundant memory capacity and the very high bandwidth data interface unit makes it possible to switch programs easily and seamlessly, even at run-time.

4 HW/SW Co-design and Verification Methodology

Figure 4 shows the HW/SW co-design and verification methodology for the 3D-SoftChip. Once HW/SW partitioning has been executed, the HW is modeled at a system level using SystemC [15] to verify functionality of the operation and to explore various architecture configurations while concurrently modeling the software in C. After this, a co-simulation and verification process is implemented to verify the operation and performance of the 3D-SoftChip architecture and to decide on an optimal HW/SW architecture at the early design stage. The rest of the procedure can be processed using any conventional HW/SW design methodology.

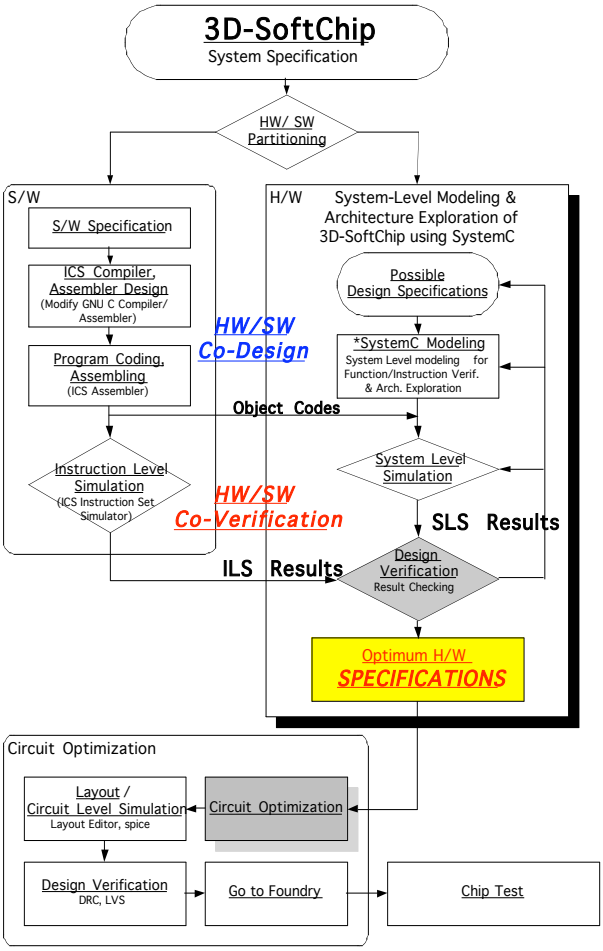
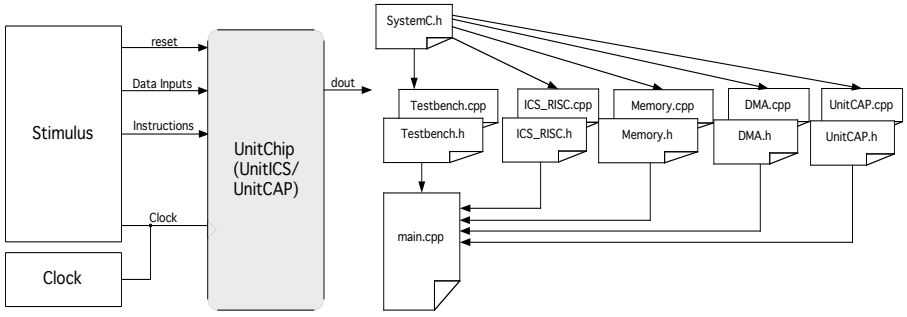


Fig. 4. Suggested HW/SW co-design and verification methodology

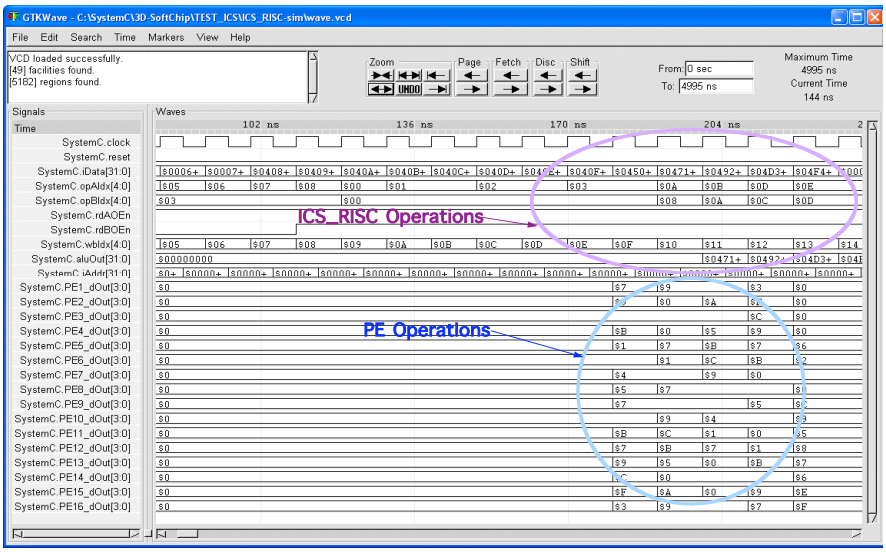
More specifically, the SW is modeled using a modified GNU C compiler and Assembler. After the compiler and assembler for ICS_RISC has been finalized, a program for the implementation of the MPEG-4 motion estimation algorithm will be developed and compiled using it. After that, object code can be produced, which can be directly used as the input stimulus for an instruction set simulator and system level simulation. The HW/SW verification process can be achieved through the comparison between the results from instruction level simulation and system level simulation. From this point on, the rest of the procedure can be processed using any conventional HW design methodology, such as full and semi-custom design.

5 High-level System Modeling and Application Mapping

The high-level system modeling has been accomplished using SystemC. A PC based development environment (Microsoft Visual C++ Version 6.0) has used to compile the high-level modelled SystemC code because of its easy accesibility. Figure 5 shows the UnitChip block diagram, SystemC file structure and the output waveform from the system-level modeling. The composition of the UnitCAP and UnitICS becomes the UnitChip. It can be largely divided into 4 kinds of sub-SystemC files, that is ICS_RISC, Memory, DMA and UnitCAP. The simple ALU instruction has been mapped in this system-level modeled UnitChip. The simulation result shows its functionality. In Figure 5(c), the upper side circle indicates the ICS_RISC operation result, and lower circle shows the PEs' operations, which is the execution of simple ALU functions in the PEs' with parallelism. The signal named as a PE1.dOut refers to the output signal from PE1. The functionality can be verified by checking these signals (from PE1~PE16) and is as expected.



(a) (b)



(c)

Fig. 5. System-level modeling of 3D-SoftChip: (a) UnitChip block diagram, (b) SystemC file structure of UnitChip and (c) the output waveform of system-level modeled UnitChip

5.1 Application Mapping for 3D-SoftChip

5.1.1 Full Search Block Matching Algorithm

Motion Estimation (ME) is introduced to exploit the temporal redundancy of video sequences and is an indispensable part of video compression standards such as the ISO/IEC, MPEG-1, MPEG-2, MPEG-4 and the CCITT, H.261/ITU-T, H.263 etc. Since ME is computationally the most demanding portion of the video encoder, it can take up to 80% of total computation time and it can be a major limiting factor for real-time performance. Among the many different ME algorithms, Full Search Block Matching (FBMA) is one of the most widely used in hardware, despite its high computational cost, because it has the optimal performance and lowest control overhead. The block matching motion estimation algorithm compares a specific sized block of pixels in the current frame with a range of equally sized pixel blocks in the previous frame to find the best match (minimum difference) between two of the blocks. The position of the best matched block can then be encoded as a motion vector for the reference block minimizing the total entropy in the frame. In FBMA the best match is determined by calculation of the sum of absolute differences (SAD) for each candidate search location *(dx, dy)* to find the minimum SAD, the SADs are calculated as follows:

$$SAD(dx,dy) = \sum_{m=x}^{x+N-1} \sum_{n=y}^{y+N-1} | I_k(m,n) - I_{k-1}(m+dx, n+dy) |$$

Where $I_k(m,n)$, $I_{k-1}(m+dx,n+dy)$ are intensity values of the pixels located at position *(m,n)* in the current and previous frame blocks respectively. In Figure 6, *(x,y)* indicates the current block pixel location, it is matched to every candidate search location within a $(2p+N-1)\times(2p+N-1)$ search window area, where *[-p, p-1]* is the pixel search range. The SAD value is calculated for every candidate block with a displacement *(dx, dy)*

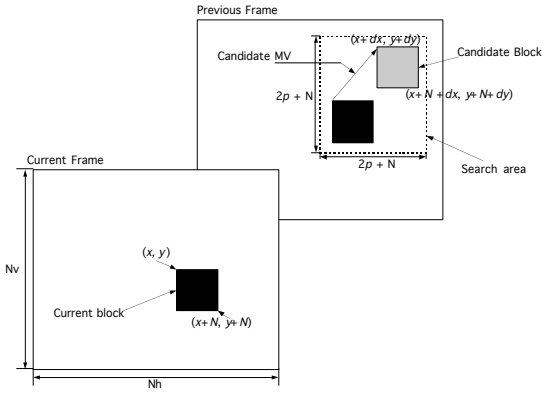


Fig. 6. Block Matching Motion Estimation

Once the SAD for each subsequent candidate block is calculated, it is compared to the existing SAD, if it is smaller than a new motion vector is stored. The calculation of SAD values and the matching process continues until all candidate blocks are matched and the overall minimum SAD is found. The stored motion vector is then the vector to the block with the best result for displacement *(dx,dy)*, which has the minimum SAD.

5.2 FBMA Mapping Method for 3D-SoftChip

Figure 7. shows the mapping method and data flow for implementation of the FBMA to the system-level modeled 3D-SoftChip. The FBMA mapping is accomplished over 10 distinct stages.

In this mapping, it is assumed the basic word-length of the S-PEs and PA-PEs is 8-bit (a simple matter of architecture scaling within each PE). The detailed explanation of this mapping is as follows:

1) STEP 1-Load REF. BLOCK DATA INTO PE ARRAY SRAM*:* The first operation is to load reference block data ($I_k(m,n)$) into embedded SRAM in each PE in the array.

2) STEP 2-EACH PE MOVES THIS DATA TO INTERNAL REGISTER: Each PE moves the reference data from the embedded SRAM into an internal register so it is available to be used for calculation of SAD values for the entire search window.

3) STEP 3-LOAD FIRST SEARCH POSITION BLOCK DATA INTO PE ARRAY SRAM: The block data for the first search position ($I_{k-1}(m+dx, n+dy)$) is then loaded into the embedded SRAM in each PE in the array ready for calculation of the SAD value between the reference block and this first search position .

4) STEP 4-EACH PE EXECUTES SUBSTRACTION AND ABSOLUTE VALUE COMPUTATION: In this step, each PE carries out a subtraction operation between the reference block data and the current search position in SRAM, the absolute value of this resulting difference is stored as the absolute difference value for that block position.

5) STEP 5-PARTIAL SUMMATION (1): In this step every odd columned PE performs a partial sum operation of its absolute difference value with the value from the PE to its immediate right in the array, the result is stored as a double-word value across both PEs.

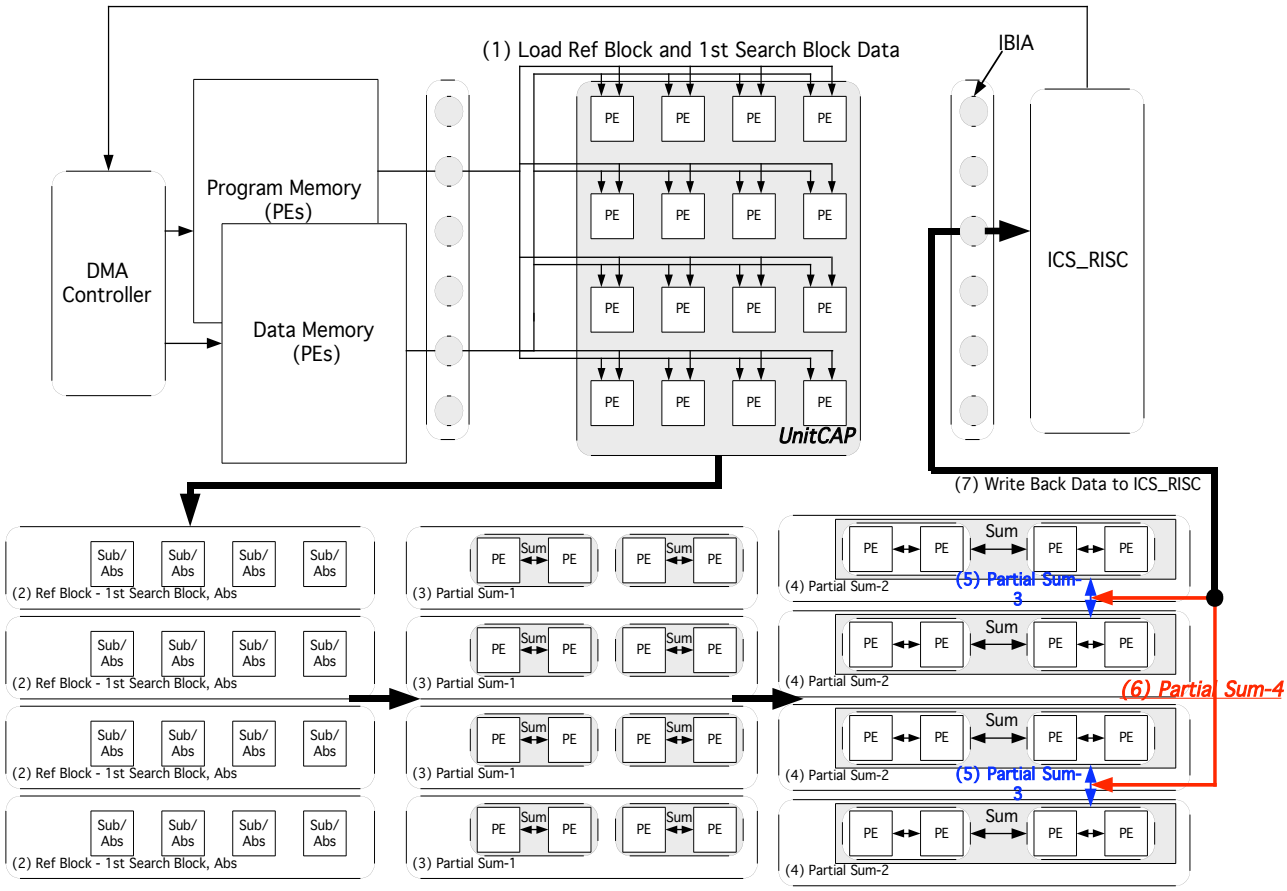


Fig. 7. Mapping Method for Full Search Block Matching and Data Flow

6) STEP 6-PARTIAL SUMMATION (2): In this step the two partial sums computed in the previous step are summed in the same way, every odd columned PE pair sums its result with the result from the PE pair to its right, this result is stored as a quad-word value across all four PEs in each row

7) STEP 7-PARTIAL SUMMATION (3): In this step the column wise operation carried out in step 5 is repeated row wise to accumulate another set of partial sums, in this case, however, the second row of PEs accumulated its result with the result from the row above, while the third row of PEs accumulates its result with the result from the row below.

8) STEP 8-PARTIAL SUMMATION (4): In this final partial sum accumulation, the second row of PEs sums its result with the result from the third row, producing the total SAD value for that search position.

9) STEP 9-WRITE BACK RESULT DATA TO THE ICS_RISC*:* Finally the resultant SAD value calculated in STEP 8 is written back to the internal register in the ICS_RISC for comparison with the previous minimum and updating of the motion vector if applicable.

10) STEP 10-REPEAT STEPS 4 TO 9*:* The next search position data block can be loaded into the SRAM in the PE array while the SAD calculation is being carried out for the current search position so once the result had been written back the calculation of the SAD for the next search position can be begun immediately.

5.3 Performance Analysis

Figure 8 shows the performance comparison of the 3D-SoftChip with a DSP processor, several ASICs and MorphoSys for matching on 8 × 8 reference block against its search area of 8 pixels displacement. There are 81 candidate blocks (27 iterations) in each search area [16]. In the 3D-SoftChip, as described above, the number of processing cycles for one candidate block is just 7 clock cycles (each UnitChip computes one quarter block, so with 4 UnitChips one complete block is computed every 7 cycles), so the total number of processing cycles for the 3D-SoftChip becomes 567 (81 iterations of 7 cycles each).

The number of clock cycles required is very close to that reported for MorphoSys, with just 4 UnitChips, this, however, can readily be improved simply by increasing the number of UnitChips on a scaled up 3D-SoftChip. A 4×4 UnitChip array, for example, would have an effective throughput of one block every 142 cycles. In addition to this, considering the characteristics of the 3D system, there are other significant advantages. Data dependency is largely eliminated so that after the initial set-up there is a 100% PE utilisation. The reference and candidate block data can be moved into the embedded SRAM in the PE concurrently with array execution, so the PEs can operate continuously. Also low power consumption can be achieved through a minimisation of the number of data accesses, because most of

data manipulation can be executed within the PE array. Most importantly, however, because all memory is directly accessible within the 3D-SoftChip via the IBIA there are effectively zero external data reads and thus power consumption will be greatly improved over all the other approaches.

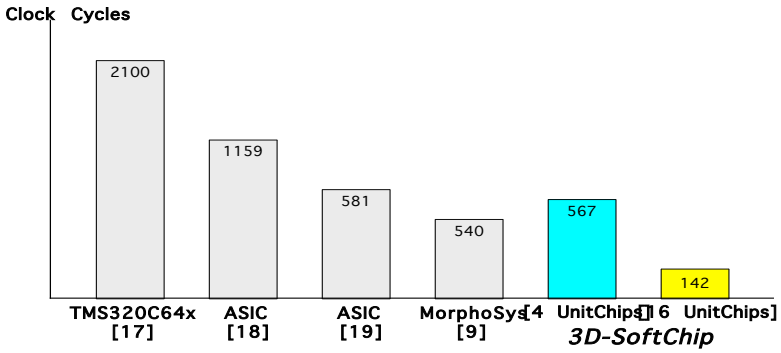


Fig. 8. Performance comparison for Motion Estimation

When comparing with the performance of the DSP processor and dedicated ASICs, the performance of the suggested 4×4 UnitChip 3D-SoftChip has remarkable advances with a theoretical capability of more than 3.8 times the performance. Given its wide applicability/adaptability to any number of other applications, the performance achieved compared to these dedicated processors is a potentially enormous advancement. This clearly demonstrates the architectural superiority of the suggested novel 3D-SoftChip.

6 High-level System Modeling and Application Mapping

The novel 3D vertically integrated adaptive system-on-chip architecture as a next generation computing system along with its functional verification and the mapping of an MPEG4 motion estimation algorithm has been presented. The performance of the execution of the MPEG full search block matching algorithm has been shown to be potentially more than 3.8 times improved over current generation processors. Due to these significant performance, power and cost advantages it can be shown that the suggested 3D-ACSoC is one of the most suitable architecture for the next generation of computing system. Moreover, the advanced HW/SW co-design and verification methodology can accelerate the reliability and significantly reduce the design time, especially the time and effort required for verification. This paper indicates a highly promising research direction for future adaptive computing systems and advanced and efficient HW/SW development methodology for ever more complicated SoCs.

7 References

1. Chul Kim et al, 3D-SoftChip: A Novel Architecture for Next Generation Adaptive Computing systems, EURASIP Journal on Applied Signal Processing, Volume 2006, Article ID 75032, (Feb. 2006), pp.1-13

2. S.Eshraghian, S.Lachowicx, K.Eshraghian, 3D-Vertically Integrated Configurable Soft-Chip with Terabit Computational Bandwidth for Image and Data Processing, Proc. MIXDES'2003, (June 2003), pp.26-28

3. J.W.Joyner, et al, Impact of three-dimensional architectures on interconnects in gigascale integration, IEEE Trans. VLSI Syst. Vol9, (Dec. 2001), pp.922-928

4. Joyner J.W, Zarkesh-Ha P.J, Meindl J.D, Global Interconnect Design in a Three-Dimensional System-on-a-Chip, IEEE Trans. on VLSI Systems, Vol. 12, Issue 4, (April 2004), pp.367-372

5. IZM, 3D System Integration;
http://www.pb.izm.fhg.de/izm/015_Programms/010_R/

6. Kaustv Banerjee, et al, 3-D ICs: A Novel Chip Design for Improving Deep-Submicrometer Interconnection, Proceedings IEEE Special Issues on Interconnections, Vol. 89, No 5, (May 2001), pp.602-633

7. C.Ebeling et al, Architecture design of reconfigurable pipelined datapaths, Advanced Research I VLSI, Proceeding 20th Anniversary Conference on (March 1999), pp.23-40, 21-40

8. E.Waingold et al, Bring it all to software: RAW Machines, Computer, Vol.30, Issue9 (Sept. 1997) pp.86-93

9. S.Hartej, L.Ming-hua, L.Guangming, J.K. Fadi, B.Nadar, M.C.F Eliseu, MorphoSys: An Integrated reconfigurable system for data-parallel and computation-intensive applications, IEEE Trans. on Computers, (May 2000), pp.456-481

10. QuickSilver Technology Inc., Adapt2400 ACM Architecture Overview;
http://ww.quicksilvertech.com/pdfs/Adapt2400_Whitepaper_0404.pdf

11. Elixent Ltd, The Reconfigurable Algorithm Processor;
http://www.elixent.com/products/white_papers.htm

12. picoChip Design Limited, PC102 Product Brief; http://www.picochip.com

13. S.Eshraghian, Implementation of Arithmetic Primitives using Truly Deep Submicro Technology (TDST), Ms. Thesis, Edith Cowan University, (2004)

14. L.Guangming, Modeling, Implementation and Scalability of the MorphoSys Dynamically Reconfigurable Computing Architecture, PhD Thesis, University of California, Irvine, (2000)

15. Open SystemC Initiative, SystemC 2.0.1 Language Reference Manual Rev. 1.0; http://www.systemc.org

16. Hartej Singh, Reconfigurable Architectures for Multimedia and Data-Parallel Application Domains, PhD Thesis, University of California, Irvine, (2000)

17. Texas Instruments, TMS320C6000 Assembly Benchmarks; http://www.ti.com/sc/docs/products/dsp/c6000/benchmarks/67x.htm

18. K.M.Yang, M-T Sun and L.Wu, A Family of VLSI Design of Motion Compensation Block Matching Algorithm, IEEE Trans. on Circuits and Systems, Vol 36, No.10, (October 1999), pp.1317-25

19. C.Hsieh and T.Lin, VLSI Architecture for Block Matching Motion Estimation Algorithm, IEEE Trans. on Circuits and Systems for Video Technology, Vol2, (June 1992), pp.167-175