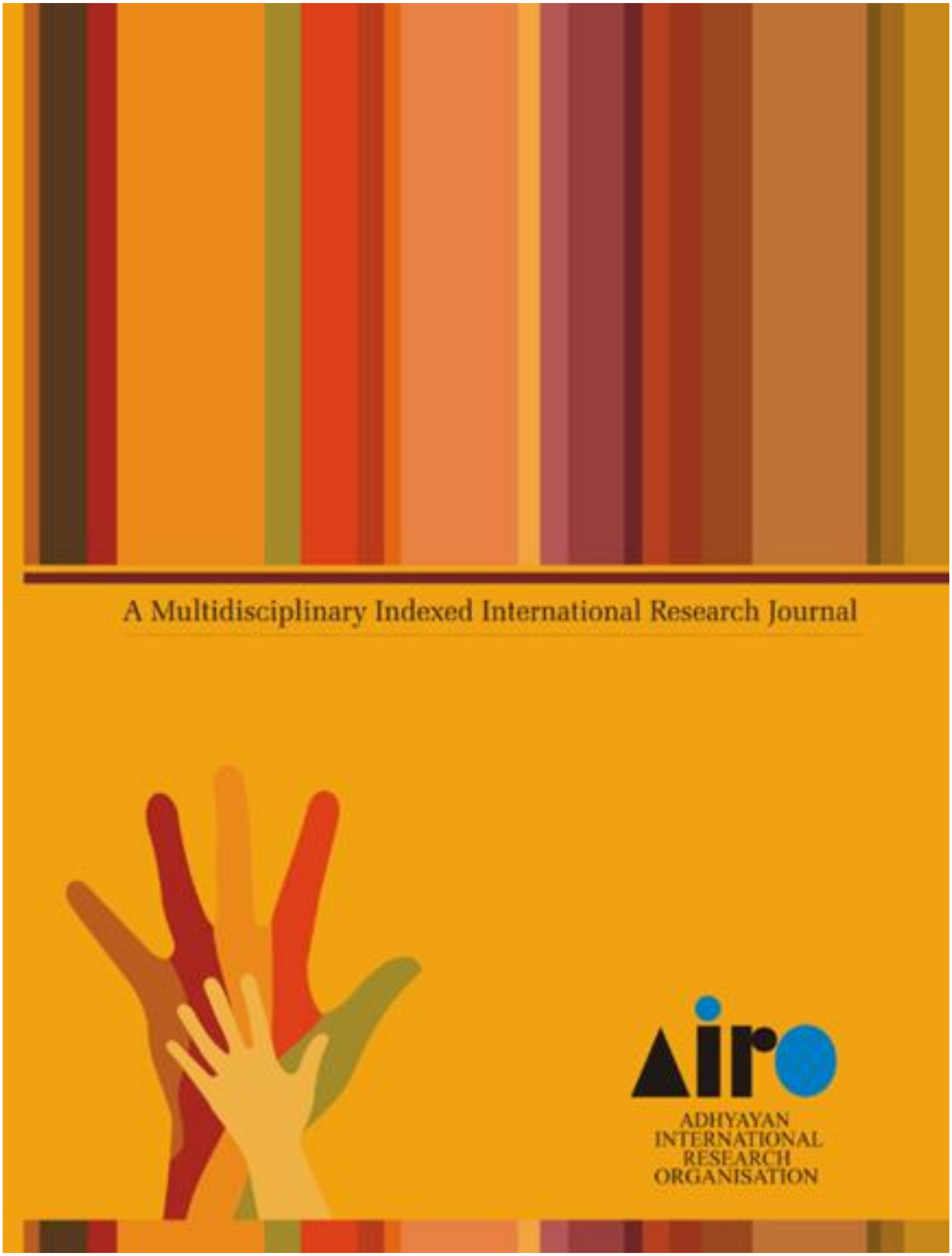
A Multidisciplinary Indexed International Research Journal
ADHYAYAN
INTERNATIONAL
RESEARCH
ORGANISATION

VLSI AND EMBEDDED SYSTEM TO EVOLUTIONARY ALGORITHMS

Acharya D. Arun Chandra
Research Scholar, OPJS University, Churu Rajasthan
Dr. Amit Jain
Assistant Professor, OPJS University, Churu Rajasthan

ABSTRACT

The complexity of VLSI and embedded systems is continually increasing and the analysis and design of such a complex system is an NP hard problem. Hardware/Software Co-Design (HSCD) is the discipline of automating the design of complex embedded systems with functionality in both hardware and software. The central task of HSCD is hardware/software partitioning which is concerned with deciding which functions are to be implemented in Hardware (HW) and which ones in Software (SW). It aims at finding an optimal trade-off between conflicting requirements on area and execution time. The problem of partitioning is also encountered in circuit layout. The layout may be generated automatically using placement and routing algorithms. The objective is to separate the cells into two or more blocks so that the number of interconnections (routing) between the blocks is minimized and the cells are evenly distributed across the layout surface. Therefore, partitioning is NP-hard and a very crucial problem.

Deterministic methods for partitioning problems provide exact solutions but when applied to complex problems they become computationally intractable. As partitioning is an NP hard problem, application of the exact methods tends to be quite slow for bigger dimensions of the problem. Heuristic/Evolutionary methods are designed for the class of problems that incorporates knowledge to itself resulting in near-optimal solutions even for larger dimensions. This research focuses on the application of evolutionary algorithms for hardware/software partitioning of embedded systems and circuit bi-partitioning for placement and routing in VLSI layout.

KEYWORDS: *VLSI and embedded systems, Hardware/Software, problem, VLSI layout, solutions, evolutionary algorithms.*

INTRODUCTION

VLSI (Very large-scale integration) is the process of incorporating or installing multiple transistors on a solitary silicon semiconductor microchip. VLSI innovation was imagined in the late 1970s when best in class level PC processor microchips were being worked on.

VLSI is a successor to small scale integration (SSI) medium-scale integration (MSI) and large-scale integration (LSI), advances.

Embedded system- Embedded system an Embedded means something that is joined to something else. An Embedded system can be thought of as a PC equipment system having programming inserted in it. An Embedded system can be an autonomous system or it can be a piece of an expansive system. It offers many advantages, for example, advanced control, accuracy timing, low unit cost, low improvement cost, high adaptability, little size, and low weight. These essential attributes can be utilized to enhance the general system or gadget in different ways:

- Improved execution
- More capacities and highlights
- Reduced cost
- Increased constancy

In light of these advantages, billions of microcontrollers are sold every year to make implanted systems for an extensive variety of items. As it were: An Embedded system is a microcontroller or chip based system which is intended to play out a particular assignment. For instance, a fire alert is an inserted system; it will detect just smoke. The electronic system which coordinates the equipment hardware with the product programming methods for giving venture arrangements is implanted systems. By utilizing this implanted system innovation the multifaceted nature of the circuits can be decreased, all things considered, which additionally diminishes the cost and size. Embedded system was fundamentally created by Charles Stark to reduce the size and weight of the task hardware.

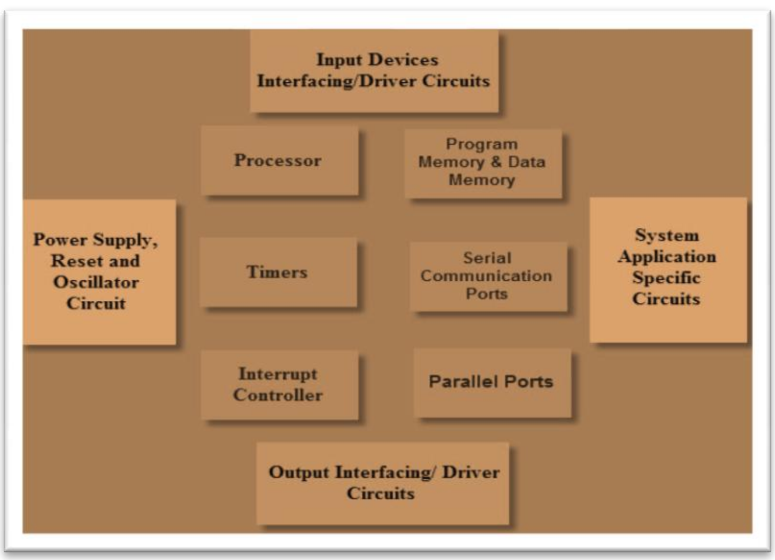

Fig: 1 Embedded Systems Design

An embedded system is mostly an electronic system that can be automate or non-programmed to operate, systematize, and execute single or multiple tasks based on the function. In the real time embedded systems, all the assemble units work together based on the program or set of rules or code embedded into the microcontroller. But, by using this microcontroller programming techniques only an inadequate range of problems can be solved.

Equipment/software co-design of embedded systems:

Inserted systems have turned into an indispensable piece of the life as buyer gadgets, auto hardware, cell phones, brilliant cards et cetera. As indicated by the worldwide innovation guide for semiconductors (Edenfled et al 2004), the most essentially tested branch of computer industry is the system plan. For a lot of usefulness of the inserted systems, the equipment and the product usage are conceivable. In such cases, the two execution alternatives normally have reciprocal preferences and burdens as sketched out in Table 1. It is helpful to utilize both equipment and programming parts to execute a given system, i.e., time-basic or power-basic segments of the system ought to be actualized in equipment, while non-basic segments ought to be actualized in programming. Therefore an ideal exchange off can be found between time, power and expenses. The ideal exchange off between the clashing parameters can be computerized utilizing the productive dividing strategies.

Parameter	Hardware	Software
Speed	Faster	Slower
Production Cost	More costly	Cheaper
Energy consumption	Lower	Higher
Heat Dissipation	Lower	Higher
Maintainability	Hard	Easier

Table 1 Hardware Vs Software Solutions

The most imperative piece of Hardware/Software Co-Design (HSCD) is apportioning, i.e., choosing which parts of the system are to be actualized in equipment and which ones in programming (Franke and Purvis, 1991). A general Computer Aided Design (CAD) structure for HSCD has five run of the mill configuration steps:

- Functional decay of abnormal state details into brings down levels.
- Partitioning the point by point detail into equipment/programming implementable sets
- Synthesis of the interface between the equipment and programming segments.
- Hardware combination
- Software combination.

The practical decay step yield is a Control and Data Flow Graph (CDFG) or Directed Acyclic Graph (DAG), which turns into the contribution to the Hardware/Software (HW/SW) apportioning step. There are two distinctive routes for system parceling in particular

(I) Structural Partitioning and

(ii) Functional Partitioning.

In basic dividing, the system is first combined and after that is parceled into squares. This sort of parceling brings about a substantial number of equipment squares. Practical apportioning separates a systems utilitarian detail into different sub-particulars. Each sub-detail speaks to the usefulness of system part, for example, a custom hardware or programming processor, and is combined down to entryways or accumulated down to machine code. Practical apportioning brings about both equipment and programming usage with less number of equipment useful pieces. The advantage of useful dividing is the request of-size lessening in rationale union runtimes (several hours down to many minutes). Useful apportioning makes it a decent strategy for application to HSCD.

REVIEW OF LITERATURE

Most of the proposed apportioning calculations are heuristic, attributable to the way that the dividing issue is NP-hard. The NP-hardness of the equipment/programming parceling was guaranteed by a few analysts (Garey and Johnson 1979, Fang and Wu 1990, Eles et al 1996, Kalavade and Lee 1994, Wolf 1994, Wolf 1997, Vahid and Gaski 1995, 2001, Binh et al 2000).

The proposed correct calculations incorporate branch-and-bound (D'Ambrosio et al 1994, Binh et al 2000), dynamic programming (Madsen et al 1997, O'Nils et al 1995, Shrivastava and Kumar 2000) and number direct programming (Arato et al 2003, Mann and Orban 2003, Niemann and Marwedel 1997).

Numerous scientists have connected universally useful heuristics to HW/SW apportioning. Especially, hereditary calculations have been widely utilized (Wu Jigang et al 2010, Wu Jigang and Thambipillai Srikanthan 2010, Dou et al 2010, Tong et al 2008, Greg Stit 2008, He Jifeng et al 2005, Zou Yi et al 2004, Theerayod Wiangtong 2004, Arato et al 2003, Rastogi et al 2001, Mei et al 2000, Dick and Jha 1997, Srinivasan et al 1998, Quan et al 1999, Lavangno et al 1994, Jantsch et al 1994, Ernst et al 1992). Reenacted Annealing (SA) was advanced by Kirkpatrick et al (1983) as another option to ravenous methodologies, which are immediately caught in neighborhood minima since they can just make downhill moves. This calculation was additionally utilized by Sudarshan Banerjee and Nikil Dutt 2004, Theerayod Wiangtong et al 2002, Eles et al 1997, Ernst et al 1992, Henkel and Ernst 2001, Lopez-Vallejo et al 2000 and Choi and Hwang 1999. Different less famous heuristics in this gathering are tabu pursuit (Theeryod Wiangtong 2004, Eles et al 1997, Lim and Chee 1991) and covetous calculations (Chatha and vemuri 2001).

A few scientists utilized custom heuristics to unravel equipment/programming dividing

(Barros et al 1993). This incorporates Global Criticality/Local Phase (GCLP) calculation (Kalavade and Lee 1993), master arrangement of Lopez Vallejo et al (1998, 1999), Lopez Vallejo and Lopez (2001) and parallel requirement look calculation by Vahid et al (1994). Concerning thesystem demonstrate, facilitate qualifications can be made. Specifically, numerous analysts consider booking as a piece of apportioning (Kalavade and Lee 1994, Yoo et al 1996, Niemann and Marwedel 1997, Balarin et al 1998, Chatha and vemuri 1998, 1999, 2000, Dick and Jha 1998, Mei et al 2000, Diessel et al 2000, Grajcar 2000, Suh et al 2000, LopezVallejo and Lopez 2001) while others don't (Eles et al 1996, Madsen et al 1997, Vahid and Lee 1997, Vahid et al 1994, 1996 and Vahid 1997). The issue of appointing correspondence occasions to joins amongst equipment and additionally programming units was incorporated by Dick and Jha 1998 and Mei et al 2000.

In various related articles, the objective engineering comprised of single programming and a solitary equipment unit (Eles et al 1996, Gupta and De Micheli 1993, Henkel and Ernst 2001, Lopez-Vallejo and Lopez 2001, Madsen et al 1997, Mei et al 2000, Srinivasan et al 1998, Greg stit and Vahid et al 2002, Vahid and Lee 1997). Different specialists don't force this impediment. A few scientists restrict parallelism inside equipment or programming (Junya Matsunaga et al 2009, Srinivasan et al 1998, Vahid and Lee 1997) or amongst equipment and programming (Henkel and Ernst 2001, Knudsen and Madsen 1998, Madsen et al 1997).

In 1970, Kernighan and Lin (KL) presented what is frequently depicted as the principal "great" diagram cut heuristic (Ouyang et al 2002). KL iteratively swaps the combine of opened modules with the most astounding addition (Gajski et al 1994, Gerez 1999). Fiduccia and Mattheyses (FM) (1982) introduced a KL-propelled calculation that lessened the time per go to direct in the extent of the net rundown. Circuit dividing utilizing tabu hunt and hereditary calculation was finished by Areibi and Vannelli (1993), Sait et al (2003, 2006).

For the min-cut division issue, Johnson et al (1989) led a broad observational investigation of reenacted tempering versus iterative change approaches, utilizing different arbitrary diagrams as a proving ground. Tabu pursuit was proposed by Glover (1989) as a general combinatorial streamlining system. The tabu rundown can be seen as a contrasting option to the securing component KL and FM. Mardhna and Ikenguchi (2003) have proposed a neuro-look based strategy for illuminating a VLSI net-list parceling issue. They clarified the key ideas of neuro-look techniques to help a VLSI net-list dividing program (Al-Abaji 2002). Hereditary dividing approach for arrangement of VLSI circuits was finished by Sipakoulis et al (1999), Srinivasan et al (2001).

In mid 1990s, Kennedy and Eberhart (1995, 1997) articulated an option answer for the complex non-straight advancement issue by imitating the aggregate conduct of flying creature runs, particles and socio-perception (Kennedy et al 2001) and called their brainchild the Particle Swarm Optimization (PSO) (Carlisle and Dozier 2000, Shi and Eberhart 2001, Bergh and Engelbrecht 2002, Brandstatter and Baumgartner 2002, Clarc and Kennedy 2002, Salman et al 2002, Wang et al 2003, Hassan et al 2004, Sudholt and Witt 2008). These calculations went for enhancement of just a single goal.

Hajela and Lin (1993) proposed a weight based hereditary calculation (WBGA) for multi-paradigm improvement (Dick and Jha 1997, Murata and Ishibuchi 1999, Jones et al 2002, Lee et al 2009). Every target work is increased by a weight and the weighted goals are added to figure the wellness of the arrangement. This technique experiences issues in discovering pareto-ideal arrangements in issues having non-curved pareto-ideal locales (Yang and Meng 2002). Vector assessed GA was proposed by Schaffer (1985).

Deb and Goldberg (2002) detailed that wellness sharing on the target work space performs superior to anything the one in view of the choice variable space. One of the burdens of the wellness sharing in view of nichie tally is that the client needs to choose another parameter σshare. To address this issue Deb and Goldberg (1989) and Fonseca and Fleming (1998) created methodical ways to deal with assess and powerfully refresh σshare. Another hindrance of niching is computational push to figure specialty tallies.

The quality pareto transformative calculation by Zitler et al (2000, 2004) utilizes a positioning strategy to allot better wellness esteems to no dominated arrangements at under spoke to districts of the goal space. As it identifies with minimization issue, the lower rank relates to better arrangements. To keep up assorted populace as required in multi-target GA, wellness sharing strategy, like Fonseca and Fleming (1998), is utilized.

Non-commanded Sorting Genetic Algorithm-II (NSGA-II) detailed by Deb et al (2002) utilizes a swarming separation methodology to have a uniform spread of arrangements along the best-known pareto-front without utilizing the wellness sharing parameter. The significant preferred standpoint of this calculation is the upkeep of elitism i.e., all non-ruled arrangements found by the calculation are held for the cutting edge by reintroducing them to the populace.

Multi-Objective PSO (MOPSO) is the original work of CoelloCoello and Lechunga (2002). In the consequent examinations (Benitez et al 2005), MOPSO was observed to be the main other option to take care of multi-target issues and can be contrasted and NSGA-II (Deb et al 2002), Pareto Archive Evolutionary Strategy (PAES) (Knowles and Corne 2000) and Micro Genetic calculation (Micro GA) (Coello-Coello and Pulido 2001).

Zitzler et al (2004) composed a pareto-streamlining agent that decides the strategy for merging to the genuine pareto-ideal front while accomplishing a very much appropriated set of arrangements. Tsou et al (2006) proposed an enhanced MOPSO with neighborhood inquiry and grouping method attempting to accomplish these objectives.

Despite the fact that the better approach to acquire an all-around circulated set of arrangements is most likely to utilize some grouping calculations that are computationally costly (Raquel and Naval 2005, Kukkonen and Deb 2006). Another changed inquiry method separated from grouping calculation was additionally distinguished. A nearby inquiry and flight component in light of packing separation is joined into MOPSO. Components in light of the swarming separation not just certainly keep up the assorted variety of the outside chronicle, yet additionally encourage the union of MOPSO-CD to the genuine pareto-ideal front (Felix.T.S.Chan and Manoj Kumar Tiwari 2007).

SYSTEM LEVEL TOOLS FOR EMBEDDED SYSTEMS DESIGN

For embedded systems, several design environments are developed such a POLIS, Ptolemy, CASTLE, COSYMA, Cobra, COSMOS, Cosyn, Coware, Chinook, Lycos, SpecSyn and VULCAN. Among them POLIS, COSYMA, Lycos, SpecSyn and VULCAN design environments are used for HW/SW partitioning. In POLIS, a design is described in a high level language (ESTEREL design Language, graphical FSMs, subsets of Verilog or VHDL) that can be directly translated into Co-design Finite State Machine (CFSMs). In order to perform hardware/software partitioning, designers interact with the system to make decisions through feedback mechanisms. POLIS is therefore not a fully automated system; it also does not have a unified design framework. In COSYMA, input to partitioning are the Extended Syntax Graph (ESG) with profiling (or control flow analysis) information. This parameter is controlled by user interaction. Partitioning works at the basic block level.

Since communication between basic blocks of a process is implicit, partitioning requires communication analysis and thereafter, communication synthesis. Communication is inserted when the internal ESG representation is translated back to C for software synthesis and to a hardware description language for high-level synthesis. LYCOS is an experimental Co-synthesis environment. In its current version LYCOS is used for hardware/software partitioning using a target architecture consisting of a single CPU and a single ASIC communicating through memory mapped I/O. It is built as a suite of tools centered around an implementation independent model of computation called Quenya, which is based on communicating control/data flow graph. Quenya directly allows design partitioning to be represented and supports representation of the system's environment in a uniform manner.

The SpecSyn system supports extensive designer interaction during all design steps. It considers all three kinds of functional objects contained in an input specification: Behaviors, variables and communication channels. During partitioning phase behaviors are partitioned over processing elements, variables over memories and communication channels over busses. The partition algorithm incorporates a hill-climbing heuristic such as simulated annealing inside a binary search procedure. VULCAN system supports hardware/software co-design of embedded systems consisting of parts that operate at different speeds, communicate with each other. Input specifications are formulated in hardware C, a C like language. As a part of specification, the designer can formulate two kinds of timing constraints: delay constraints and execution rate constraints. The synthesized system will be implemented on an architecture consisting of one processor, one hardware component (ASIC), one system bus and one global memory through which all hardware/software communication takes place.

Table 2 summarizes the main focus of existing design environments that are developed for embedded system design (Theerayod Wiangtong, 2004). All these systems are based on embedded systems suitable for control oriented applications, or systems consisting of processors and ASICs.

Table 2- Tools used for Embedded System Design

Tools	University/Developer	Main focus
CASTLE	SET	Design space exploration
Chinook	U Washington	Interfacing
Cobra	U Tubingen	Prototyping
COSMOS	TIMA	Refinement
COSYMA	TU Braunschweig	Partitioning
Cosyn	Princeton	Co-synthesis
Co Ware	IMEC	Interfacing
LYCOS	TU Denmark	Partitioning
POLIS	UC Berkeley	Partitioning, Verification
Ptolemy	UC Berkeley	Modeling, Simulation
SpecSyn	UC Irvine	Specification, Partitioning
VULCAN	Stanford	Partitioning

CONCLUSION

The complexity of VLSI and the embedded systems is continually increasing and the analysis of complex systems becomes an NP hard problem. The development of efficient partitioning methodologies has been an active area of research, since early 1990s. The difference between hardware and software as well as their interaction contributes significantly to the huge complexity of the systems. EDA tools have become a necessity to automate and optimize the design process and produce good quality design in a reasonable amount of time. One of the most crucial steps in the design of embedded systems is HW/SW partitioning, i.e. deciding which components of the system are to be implemented in hardware, and which ones in software. This work focuses on the application of evolutionary algorithm for HW/SW partitioning of embedded systems and circuit bi-partitioning for placement and routing.

Evolutionary algorithms are powerful search techniques that are used to solve complex optimization problems. Optimization refers to finding one or more feasible solutions that corresponds to the extreme values of one or more objectives. The multiobjective optimization algorithms discussed in this thesis are used for optimizing two objectives namely the area and the execution time. These algorithms can be extended for optimizing more number of objectives (area, power, latency, execution time) simultaneously.

REFERENCES:

[1] Al-Abaji, R. H. "Evolutionary techniques for multi-objective VLSI net list partitioning", Master's Thesis, King Fahd University of Petroleum and Minerals, Dhahran, Kingdom of Saudi Arabia, 2002.

[2] Alvarez-Benitez, J. E., Everson, R. M. and Fieldsend, J. E. "A MOPSO algorithm based exclusively on pareto dominance concepts", Lecture Notes in Computer Science, 3410, pp. 459-473, 2005.

[3] Andreatta, A. A. and Ribeiro, C. C. "A graph partitioning heuristic for the parallel pseudo-exhaustive logical test of VLSI combinational circuits", Annals of Operations Research, pp. 1-36, 1994.

[4] Arato, P., Juhasz, S., Mann, Z. A., Orban, A. and Papp, D. "Hardware/ software partitioning in embedded system design", in Proc. of the IEEE International Symposium on Intelligent Signal Processing, 2003.

[5] Arato, P., Mann, Z. A. and Orban, A. "Genetic scheduling algorithm for high-level synthesis", in Proc. IEEE 6th International Conference on Intelligent Engineering Systems, 2002.

[6] Areibi, S. and Vannelli, A. "Circuit partitioning using a tabu search approach", in Proc. of IEEE International. Symposium on Circuits and Systems, pp. 1643-1646, 1993.

[7] Arunachalam, V., Sapra, S., Chaitanya, N. S. K. and Prakash Rainac, J. "Hardware/software partitioning algorithm for embedded systems with repeated functionalities", in Proc. of TENCON 2008 IEEE Region 10 Conference, pp. 1-6, 2008.

[8] Bakshi, S. and Gajski, D. "Partitioning and Pipelining for Performance- Constrained Hardware/Software System", IEEE transaction on Very Large Scale Integration Systems, Vol. 7, No. 4, pp. 419-432, 1999.

[9] Bergh, F. and Engelbrecht, A. “A new locally convergent particle swarm optimizer”, in Proc. of Conference on Systems, Man and Cybernetics, pp. 96-101, 2002.

[10] Binh, N. N., Imai, M., Shiomi, A. and Hikichi, N. “A hardware/ software partitioning algorithm for designing pipelined ASIPs with least gate counts”, in Proc. of the 33rd Annual Conference on Design Automation, pp. 527-532, 2000.

[11] Blickle, T. “Theory of evolutionary algorithms and applications to system Synthesis”, Ph.D. thesis, Swiss Federal Institute of Technology, Zurich, 1996.

[12] Brandstatter, B. and Baumgartner, U. “Particle swarm optimization – Mass-spring system analog”, IEEE Transactions on Magnetics, Vol. 38, pp. 997-1000, 2002.