Design Planning for Single Chip Implementation of Digital Wireless Mobile Transceiver

Vineet Sahula
Department of Electronics & Comm. Engineering
Regional Engineering College
Jaipur, India - 302 017.
sahula@ieee.org

C. P. Ravikumar
Department of Electrical Engineering
Indian Institute of Technology
New Delhi, India - 110 016.
rkumar@ee.iitd.ernet.in

Abstract— **In this paper we present a design planning paradigm for design of a wireless mobile transceiver. We consider digital part in single chip implementation of a transceiver based on CDMA spread spectrum technique. The complexity of such a chip implementation makes the design process, complex and very expensive. The desired characteristics of a mobile transceiver are low cost, small size, and low power. Design cost forms a major portion of total system cost. In order to reduce design cost, design completion time should be reduced. We assume hardware-software design flow for design of the transceiver. We analyze the design flow using the hierarchical concurrent flow graph (HCFG) approach. We illustrate, using AND and OR concurrent constructs of HCFG approach, how the design process completion time can be reduced by employing concurrent design efforts. We also present an approach for completion time improvement which considers sensitivity of completion time with respect to task completion time and probabilities. HCFG analysis facilitates a pre-execution *"what-if"* analysis to determine suitable design flow which provides lowest process completion time.**

I. Introduction

With the growing popularity of mobile wireless communication, the demand for high-fidelity, low-power, and low-cost end-user systems has increased dramatically. Very Large Scale Integration forms the basis for the implementation of such systems. While performance, cost, and power dissipation are undoubtedly the key issues on which two competing products will be compared, the time to market the product will decide the survival of the product in the market. The first product to reach the market will take away a giant share in the sales, as is evident from marketing surveys. It is therefore clear that the design of any commodity product in a popular area of electronics is geared towards cutting down the time-to-market. Design time forms a major part of the time-to-market for complex electronic systems-on-chip which are used in the construction of mobile end-user systems. In this paper, we shall consider the optimization of an electronic design process meant for designing the digital part of a mobile wireless transceiver. We shall use a representation called Hierarchical Concurrent Flow Graph (HCFG) [6] for capturing the details of the design process and for estimating the design time.

We study transformations to the design process in order to improve the estimated design time. These transformations include

- Splitting a design task that is time consuming and introducing concurrency in the execution flow
- Using alternate subflows to complete a design task

We have built a tool which can estimate the design completion time for a given design process, and evaluate user-suggested transformations to the design process to reduce the completion time. While our ultimate aim is to automate the optimization process, we have found that a user of the tool can gain insight into the bottlenecks in the design process and perform a "what-if" analysis with the help of the tool to explore the design flow options. Coupled with a grapical user interface, our tool can be invaluable to design managers.

We have organized the paper as follows. In Section II, we explain the HCFG approach to model design processes. Section III outlines a possible design for a transceiver implementation, which we shall use as an example to illustrate our estimation method. We shall consider a hardware-software codesign flow based on the Ptolemy approach [5] to design the transceiver. Section IV shows the application of HCFG technique to the Ptolemy design flow presented in Section III. Process graph transformations are presented in Section V. Conclusions are given in Section VI.

II. Brief overview of HCFG approach

A. Design flow graph

A design process can be represented as a weighted directed graph, whose nodes represent design tasks and directed edges represent task sequencing information. The weight T_j associated with a node is a discrete random variable which represents the completion time of task j. Let $E[T_j]$ denote the expected value of T_j. $Mode[T_j]$ is the most likely value that T_j takes on. The weight p_{ij} on an edge (i, j) in the graph represents the probability that task j will be performed immediately after i. When there is no concurrency in the design flow, the sum of the probabilities an all outgoing edges from a node is 1. One is interested in the following quantities (a)expected value $E[T_P]$ and (b)Most likely value $Mode[T_P]$ for the completion time of entire design.

This work was supported by a research grant from Motorola.

B. AND and OR concurrent constructs

The VLSI design process is hierarchical. In our graph representation of design flow, we shall assume that a node represents either a design task or a subflow. Also there is considerable amount of concurrency in a VLSI design flow. Smaller teams of designers may be working on different parts of a system concurrently. We refer to such a form of concurrency as AND-concurrency. The subflows related by the AND-concurrency are enclosed between two "operation-nodes" denoted by $\odot$. The expected completion time for the complete task in Figure 1, is

$$E[T_P] = E[T_{DP}] + E[Max\ \{T_{ALUD}, T_{MD}, T_{CD}\}] + E[T_{DI}] \quad (1)$$

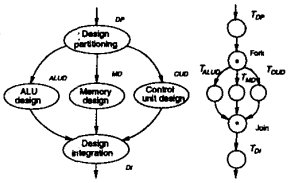

Fig. 1. Example design flow with AND-concurrency

Another form of concurrency is OR-concurrency, where many design teams attempt to solve the same problem through alternate techniques. Two OR-operation nodes are used to enclose the subflows related by OR-concurrency in the design flow graph. These nodes are labelled $\oplus$. The operation nodes are not associated with node weights.

In the example of Figure 2, the expected completion time is

$$E[T_P] = E[Min\ \{T_{L_1}, T_{L_2}, T_{L_3}\}] \quad (2)$$

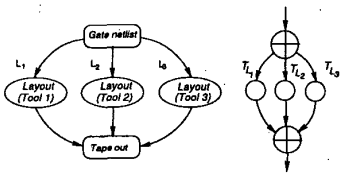

Fig. 2. Example design flow with OR-concurrency

C. Constructing equivalent HCFG

From the design flow graph $\widehat{G}$, we construct a hierarchical concurrent signal flow graph G as follows. We add two extra nodes I and F that represent the *initial* and *final* tasks, both with zero task completion time. We draw directed edges of the form (I, x) in G, where x is a node in $\widehat{G}$ with zero in-degree. Similarly, we include directed edges of the form (y, F) in G, where y is a node in $\widehat{G}$ that has zero out degree. For every edge (i, j) in $\widehat{G}$, the corresponding edge in G is associated with an edge transmittance $\mathcal{T}_{ij} = p_{ij} \cdot z^{T_j}$. The transmittance of an edge (I, x) is z^{T_x} and the transmittance of an edge (y, F) is 1. Here z represents the transform variable. As per convention T_j is treated as an integer.

Mason's Gain formula can be used for calculating [2]. Given $\mathcal{T}_{i,f}$ it can be shown that

$$E[T_P] = \left.\frac{d\mathcal{T}_{i,f}}{dz}\right|_{z=1} \quad (3)$$

Similarly, the impulse response of $\mathcal{T}_{i,f}(z)$ represents the PDF of the completion time T_P.

III. Flow for wireless mobile transceiver design

We now consider the design flow for a wireless digital transceiver used in an end-user unit. A single chip implementation of the transceiver requires mixed signal design techniques and hardware-software codesign methodology [1]. The RF section in the transceiver is realized using analog ASICs, whereas the digital baseband is realized using DSPs and digital ASICs [3]. A single chip implementation using ASICs, a DSP and a microcontroller is shown in Figure 4. The choice of hardware-software codesign methodology such as Ptolemy [5] is natural for the baseband section of wireless transceiver, as it facilitates the co-verification of hardware and software modules. We model the Ptolemy codesign flow for a wireless transceiver as shown in Figure 5.

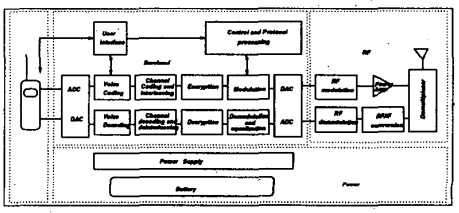

Fig. 3. Generic schematic of wireless mobile transceiver

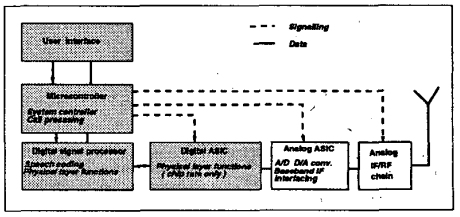

Fig. 4. Chip block diagram for baseband functionality of transceiver of Figure 3

The Ptolemy approach to the development of the digital portion of a wireless transceiver can be summarized as follows. The process begins with a complete functional specification of the transceiver. Algorithms are selected for various blocks such as channel coding, speech coding etc. Since implementing the entire transceiver in hardware is too costly, and entirely in software is too slow,

there is a need to perform hardware-software partitioning. Thus keeping the cost and performance tradeoff in mind, the functionalities are partitioned into hardware and software blocks. The hardware and software elements of the transceiver are concurrently designed. We refer to this aspect as inherent concurrency where the hardware and software development tasks are actually subflows. A coverification step is used to check the correctness of the design. Iterations which lead us back to algorithm development stage are possible when coverification indicates that the system fails to meet the specifications.

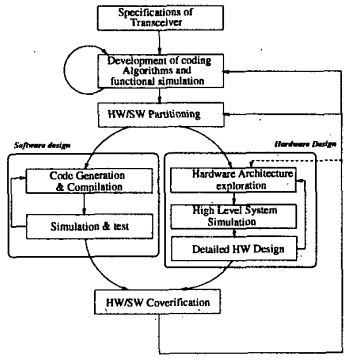

Fig. 5. Ptolemy hardware-software codesign methodology

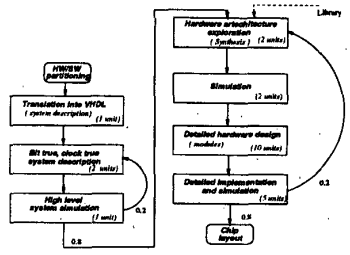

Fig. 6. Hardware design

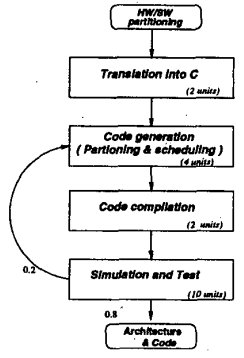

Fig. 7. Software design

A. *Hardware and software design flows*

DSP algorithms used to realize different functionalities in the transceiver can either be implemented in hardware using ASICs, or in software using a programmable DSP [8][7]. Considerations for hardware implementation include precision, clocking, resource sharing etc. Hardware design begins with a behavioral-level specification and goes through tasks such as behavioral-level description of the system using an HDL, RTL synthesis, and detailed physical design as shown in Figure 6. Considerations in software implementation include real-time operation, scheduling, and interprocessor communication. The algorithm is described and simulated in a high level language (C/C++), as depicted in Figure 7. For implementation on DSPs, simulated code is translated and compiled into DSP processor-specific code. For simulation of compiled code, the DSP itself may be used as simulation platform.

IV. HCFG REPRESENTATION

During project planning, we are concerned with time planning and resource allocation for the project. Each concurrent task in a flow requires separate resources. We assume that unlimited computing machines and design tools are available. We can vary the allocation of man-power resources to the flow and explore tradeoffs between process completion time and effort with the object of minimizing process completion time.

The inputs to the HCFG model for the example of wireless transceiver design are shown in Table I. We consider the following design options: (1) Entirely sequential execution of the flow, (2) Exploiting inherent concurrency, and (3) Introducing AND and OR concurrency in HW and SW design.

The flow for option 1 requires only one design group, with only one of the designers being active at any time. Flow of option 2 employs two expert groups, one in software design and another in hardware design, both the groups being active concurrently. This further implies that *Detailed HW design* task for n-modules would be performed sequentially i.e. one module design at a time. For flow of option 3 assuming that unlimited man-power resources are available, we propose following. (1)For SW part of design subflow, we explore the available library of processor architectures and select the most suitable DSP. This design task can be assigned to two or more software design groups. Thus multiple groups will work concurrently on redundant subflows for SW design and the software code to become available earliest will be treated as solution. (2)For HW design subflow we assign design task *Detailed HW design* to multiple designers each of them working concurrently on design of different modules. After the introduction of additional AND and OR concurrent constructs, the HCFG equivalent for the flow of option 3 is shown in Figure 8.

The results of the HCFG analysis for this example are shown in Table II. We see that option 2 gives a considerable improvement in $E[T_P]$ in comparison to option 1. Option 3 is resource-intensive, but does provide further reduction in completion time. We also compute the range of design

TABLE I
HCFG model input parameters

Task Notation	Description of task	Atomic/ subflow	Task completion time	
			Attribute	Value/Range (weeks)/Transmittance
SP	Specification	Atomic	Uniform Random Variable	2-4
AD	Algorithm Development	Atomic	Uniform Random Variable	3-7
Part	HW/SW Partitioning	Atomic	Constant	1
SWD	Software Design	Subflow	PDF	$0.7z^{18} + 0.21z^{33} + 0.06z^{48} + ...$
HWD	Hardware Design	Subflow	PDF	$0.72z^{26} + 0.07z^{29} + 0.007z^{32} + ...$
CV	Coverification	Atomic	Uniform Random Variable	2-6

TABLE II
Comparing results of three options for wireless transceiver design flow

Parameters (All time values are weeks)		HCFG Analysis		
		Option 1 Pure sequential execution	Option 2 Execution with inherent concurrency	Option 3 Execution with additional concurrency
Expected completion time, $E[T_P]$		79	52	46
Standard Deviation, σ_{T_P}		31	19	18
Most probable completion time, $MODE[T_P]$		59	41	35
Probability of, $MODE[T_P]$		0.043	0.044	0.06
Confidence interval	90 %	55-278	37-179	30-192
	99 %	52-278	34-179	28-192
Expected Completion Effort (Weeks)		79	79	106

time for 90% and 99% chances of completion.

From the PDF of process completion time and effort, it is possible to answer questions such as, (1)What is chance that using option 3 will ensure at least 25% improvement in expected completion time as compared to option 1. In this example, the answer to the question is 0.4. (2)What is the chance that flow of option 3 will ensure not using more than 2 times the resources as compared to option 1. In this example, the answer to the question is 0.92. (3)What is chance that option 3 will ensure not using more than 1.33 times the resources as compared to option 1. The answer to this is 0.77. Tables III and IV illustrate such conclusions.

TABLE III
Completion time comparison

How much better completion time is ?		Chances comparing option 3 with	
		Option 2	Option 1
$\frac{T_{P_\#} - T_{P_3}}{T_{P_\#}}$	$\geq$10%	29%	45%
	$\geq$25%	16%	40%
	$\geq$50%	5%	17%
$T_{P_3} > T_{P_\#}$		16%	6%

corresponds to either 1 or 2 depending upon the option

In the section V, we illustrate how the process completion time can be further improved.

TABLE IV
Completion effort comparison

How much extra efforts are required?		Chances comparing option 3 with	
		Option 2	Option 1
$\frac{Eff_{Option_3}}{Eff_{Option_\#}}$	$\geq \frac{10}{9}$	33%	34%
	$\geq \frac{4}{3}$	22%	23%
	≥ 2	8%	8%

corresponds to either 1 or 2 depending upon the context of chosen option

V. Design Completion time improvement

Process bottlenecks has to be identified in order to improve completion time of a process. Rigorous sensitivity analysis can be performed to obtain process completion time sensitivities with respect to task times and iteration probabilities. Although we exclude sensitivity analysis from present purview of discussion yet we depend on the guidance provided by the analysis in identifying process bottlenecks. Following may be the major contributing factors to process completion time. (1)Iteration probabilities of large values (2)Iteration probabilities of moderate value but enclosing larger size loop (3)Tasks with large completion times alone or enclosed in an iteration loop. For example Figure 10 illustrates the effect of task time variations and iteration probability variations on process completion time. We choose task times T_{AD} and T_{HWD} as variables. We also assume that iteration probability p from *Coverifi-*

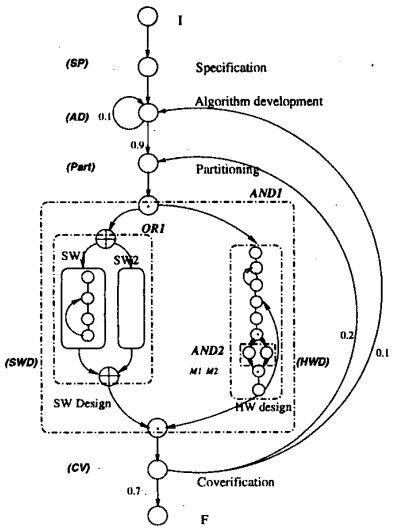

Fig. 8. HCFG representation of wireless transceiver design flow

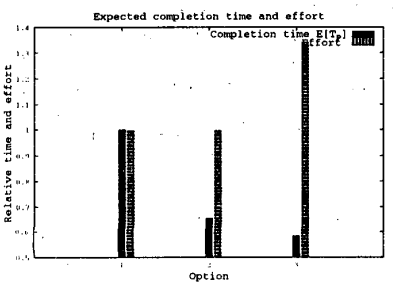

Fig. 9. Comparison of process completion time and effort for three options

cation to *Algorithm development* is also dependent on task time T_{AD} as follow.

$$p = p_{coverification} \cdot B^{-T_{AD}} \cdot \frac{1+T_{AD}}{T_{AD}} \tag{4}$$

where B is a constant and $p_{coverification}$ is original probability in the flow. We justify the form of the equation 4 on the the basis of the intuitive expressions, $\lim_{T_{AD}\to 0} p \to \infty$ and $\lim_{T_{AD}\to\infty} p = 0$.

We use following relations while estimating process completion time $E[T_P]$.

$$E[T_{AND1}] = E[Max\{T_{SWD}, T_{HWD}\}] \tag{5}$$
$$\cong Max\{E[T_{SWD}], E[T_{HWD}]\} \tag{6}$$

$$E[T_P] = \frac{T_{SP} + T_{AD} + E[T_{AND1}] + T_{CV}}{1-p} \tag{7}$$

For the above form of dependence in equations 4-7, the completion time variation is shown in Figure 10. We observe that optimal task time for Algorithm Development is 4 weeks which results in smallest process completion time of 34.6 weeks. Similar analysis can be performed for other iteration probability loops and optimal values of task times can be computed which result in lowest $E[T_P]$.

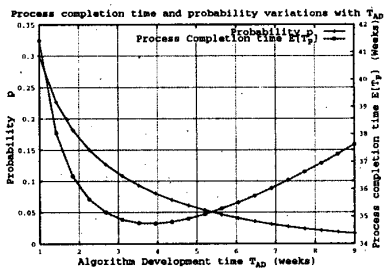

Fig. 10. Process completion time variation with T_{AD}

VI. Conclusions

We have applied the HCFG approach for analysis of the design flow for wireless transceiver. With increasing advancement in networking and wireless technology, growth in size and functionality of communication systems has been tremendous. HCFG approach is suitable for analysis of complex design processes for such large systems. We have illustrated that use of concurrent construct can be made to improve process completion time. There is a corresponding penalty involved in using concurrent constructs. For the use of OR construct, penalty is in the form of increased design effort whereas use of AND construct implies penalty in the form of decreased utilization factor.

References

[1] E. Bolsens et al. Hardware software codesign of digital communication systems. *Proceedings of IEEE*, 85(3):397–418, Mar. 1997.

[2] W. K. Chen. *Applied Graph theory*. North Holland pub. Co., 1971.

[3] Z. Kostic and S. Seetharaman. DSPs in cellular radio communication. *IEEE Communications Magazine*, pages 22–35, Dec. 1997.

[4] L. Philips, I. Bolsens, and H. De Man. A programmable CDMA IF transceiver ASIC for wireless communication. In *Proceedings of IEEE Custom Integrated Circuits Conference*, 1995.

[5] Z. Rozenblit and K. Buchenrieder. *Codesign : Computer aided software hardware engineering*. IEEE Press, 1995.

[6] V. Sahula and C. P. Ravikumar. Hieararchical Concurrent Flow Graph approach for modeling and analysis of concurrent design processes. In *Proceedings of* 14^{th} *IEEE International Conference on VLSI Design*, To appear in 2001.

[7] K. S. Shanmugam. Simulations and implementation tools for signal processing and communication systems. *IEEE Communications Magazine*, pages 36–40, Jul. 1994.

[8] J. H. Snyder et al. Tools for real time DSP research. *IEEE Communications Magazine*, pages 64–74, Nov. 1993.