

Managing Risk in Block Based Designs: A Front End Acceptance Methodology

Kumar Venkatramani and Stefanus Mantik
Cadence Design Systems, Inc

Abstract

It has recently become commonly accepted knowledge that System on Chip designs are tending to use more and more pre-built IP blocks as the most common way to achieve more reuse and faster time to markets [1]. However, as most designers who have attempted this approach, will substantiate, this approach has lots of pitfalls and unforeseen risks. In this paper, we outline a very formal, step-by-step process or methodology by which design specifications as well as component blocks are analyzed for correctness, completeness and feasibility even before the design is started. This process is used as a way to quantify risk very early in the design phase. Having analyzed the data and understood the risks, design teams will be better suited to improve predictability. This formal methodology also reduces the dependencies on the constant availability of experts and allows design teams to supplant their individual expertise with knowledge of the collective.

1. Introduction

Advances in semiconductor technologies have made significant more chip area as well as faster clock speeds available for design starts. Three forces counter the natural desire to integrate more and more functions on the same chip. First, significant challenges exist in understanding, modeling and dealing with the smaller geometries at which these semiconductors operate. For example, the signal integrity effects are 0.13 micron and below have significant impact on the yield. Secondly, simplifying assumptions made in the past ten years are no longer valid at today geometries. For example, wire delay can no longer be considered insignificant compared to gate delays. Thirdly, crucial EDA algorithms that were NP complete are taking longer and longer to compute as the number of elements they are operating on are growing significantly. Verification of devices in conjunction with the number of possible input-output combinations grows exponentially as devices (and their inputs and outputs) grow. Hence it is becoming more and more difficult to fill the available silicon with working devices. This has been widely publicized as the design productivity gap.

These complex requirements are pushing chip designers to find new and innovative ways to overcome the gap. One methodology that is gaining popularity is the *block-based design (BBD)* methodology where a new chip comprises several pre-built and pre-verified intellectual property (IP) blocks that provide the desired functions thereby reducing the number of devices or elements that must be processed as part of a new design start.

While BBD methodology gives some hope in reducing the design cycle time by using predefined IP blocks, there are some associated risks that need to be considered. First, each IP block has its own design requirements that may not be compatible with the customer specification or with other blocks that have been selected for the design. For example, a peripheral IP block must be built to communicate with the same bus as all other blocks in the design. Secondly, IP blocks may show problems when applied to a system, which was not obvious, when tested stand-alone. Thirdly, IP blocks from different vendors might come with incompatible test strategies. Lastly, if there are more than one blocks that match the specification, some quality assessment need to be performed to guide the selection process.

Most BBD approaches begin with an obvious selection of blocks, based on internal availability or existing relationships with vendors. Additional blocks, either unique to the design or the specifications, are also added to the mix. Finally gaps in the design are then filled with third party IP. Since most designs have something unique in terms of either critical design parameters or new process issues, predicting design time becomes difficult or people have to make assumptions about the feasibility of the design in the time frames specified. This process, however, can be unpredictable and relies heavily on the individual team that is assigned to complete the design. Another possible approach is to select alternative available blocks and process these blocks in parallel. Unfortunately, this approach requires a large number of resources (e.g., machines, licenses, software, people etc.).

In this paper, we outline a very formal, step-by-step process or methodology called Front End Acceptance in which design specifications as well as component

blocks are analyzed for correctness, completeness and feasibility even before the design is started. This process is used as a way to quantify risk very early in the design phase. Having analyzed the data and understood the risks, design teams will be better suited to improve predictability. This allows design teams to supplant their individual expertise. Front-end acceptance process will estimate the potential risk associated with each block and is used to predict the design budgets and plans.

2. Front-End Acceptance Methodology

Front-end acceptance (FEA) is a methodology for the analysis of a proposed chip design consisting of various functional blocks. This analysis enables a chip design team to determine the integrity of design data, evaluate associated design risks, and develop design budgets and plans. FEA is used as the starting point of the block-based design process. It provides (i) alignment of expectations between the customer and chip design team (ii) agreement on executable specification between the customer and the design team, (iii) increases in predictability of design costs, schedules, and critical design parameters. (iv) minimization of "downstream" development costs, (v) reduction in the risk of not meeting customer specifications and subsequent design cycle reiterations, and (vi) facilitation for efficient design data management.

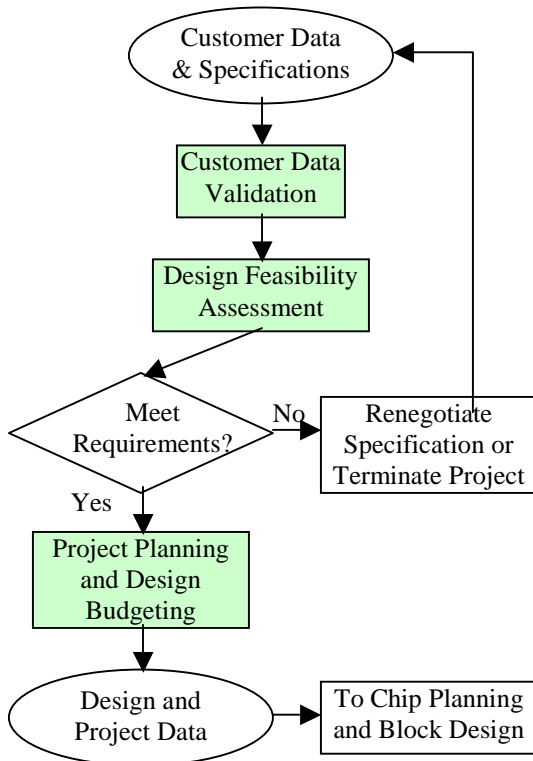


Figure 1: Front End Acceptance flow.

The Front End Acceptance methodology is broken up into three major parts, namely, Customer Data Validation, Design Feasibility Assessment as outlined in Figure 1. Each step is described in the following sections.

3. FEA Preparation Step

Two factors must be completely characterized prior to executing this methodology. They are Field Of Use and Field of Experience.

The Field of Use defines the design inputs, characteristics and process parameters that are supported by a given methodology. When a methodology is applied within that field of use, the result is rapid, predictable design implementation. The result is likely to be compromised if the methodology is applied outside the field of use.

Every design group is characterized by their expertise and the experiences that have guided them in the past. Most often, however, this information is typically not written down and is used in a very ad-hoc manner. These experiences are labeled as Field of Experience in context of this methodology. *The Field of Experience is the body of knowledge concerning designs previously developed by a design team.* The construction of a field of experience with block classifications and subclasses is detailed in a separate white paper [2].

4. Customer Data Validation

The first step of the FEA is the validation of customer data. The validation process involves the formal handoff and ensuring the correctness and completeness of design files and requirements including constraints, implementation specifications, and product objectives and goals from a system design to a chip design team. This step can be broken down further into four sub-processes: (i) project data management, (ii) FEA design review, (iii) block selection and (iv) customer data checking.

4.1. Project Data Management

In this step, a directory structure is created to store all necessary files that are related to this project. The customer design data are moved into this directory. Also, a configuration management process for handling design data is established. It manages the revision of the data as they evolve through the BBD development process. It also manages design abstractions. This step is critical to track design data handoffs between the system design team and the chip implementation team. This directory structure scheme provides a consistent

method to store information such as source files, testbenches, configuration files, golden simulation test vectors, and design libraries. The use of the configuration management process also controls designer access to the most recent, tested and verified design data.

4.2. FEA Design Review

In the second step of the customer data validation phase, the customer data are reviewed. Three separate steps are executed as part of this process, namely, scope verification relative to field of use, checklist execution, where-in the design data is run against a pre-established check list to ensure completeness and correctness of the design data and finally, Design for Test (DFT) testability checks.

In the scope verification step, design specifications and preliminary block selections are checked against the BDD field of use where the design team can determine whether or not the design is truly a block-based design and can be developed using the BDD methodology.

Once the design has been tested for its suitability with BDD methodology, the design team needs to check the completeness of design data that has been received from the customer. These data include (i) chip requirements (area, cost, performance, process technology, cell libraries, schedule, power, manufacturability, signal integrity, reliability, quality), (ii) package selection, (including die size, pin out, reliability, bondability checkers, qualification, cost, electrical characterization and mechanical characterizations) and chip level specifications and requirements (including block diagrams, bandwidth requirements, protocols, shared memory requirements, clocking system requirements, glue logic requirements, chip level testbenches, golden reference signals, data flow simulations, and analog/mixed signal requirements. If any data is missing or incomplete, the design team collaborates with the customer to acquire the necessary data.

Lastly, the design team performs checks to ensure that high-quality manufacturing test suite can be developed for the proposed design, and that this suite meets the customer requirements for test coverage, test suite size, and test suite run time. The types of tests performed include testability (scanability) of soft blocks, test coverage for virtual component (VC) test suites, and collaring requirements.

4.3. Block Selection

If the customer does not specify the blocks to be used in the design, the design team needs to select blocks before continuing the FEA process. In this step, the block

information is obtained for design estimation. All non-feasible blocks as well as non-optimal blocks are eliminated.

For each block, its suitability and applicability to meet requirements for architecture, power, DFT techniques, timing, clock structure, architecture, area, and verification must be studied from the claims provided by the block. It is a good idea to check the block's documentation against the VC documentation checklist identified in the VC Transfer document of VSIA [3].

In critical blocks, it is further recommended to verify the claims and assumptions of the block, by running functional simulations, running power analysis tools, running static timing analyzer tools, and verifying the layout (for hard blocks) by loading the appropriate abstracts into the floor planning tools. The tools to be used during this step are the tools that will be used as part of the design implementation not the tool that we originally used to design the block.

When blocks don't meet the requirements, a risk, effort, and cost matrix must be developed to select the appropriate choice of whether to accept and modify the block, or re implement the block from scratch. The effort column must be evaluated against criteria such as source code availability, amount of documentation and level of documentation available, test examples available, use of parameterization, ability to synchronize the inputs and outputs, ability to apply different DFT techniques, ability to talk to the bus of choice for the design, ability to vary the timing of the block to meet setup and hold requirements.

4.4. Customer Data Checking

The last step in this phase is to check the supplied design information from customers. There are several forms of information that the customer can supply: (i) ASCII text or paper format such as datasheets, block diagrams, descriptions, etc., (ii) data files such as constraint files, testbenches, and netlists, or (iii) executable files such as RTL code and functional models. The design team validates this incoming data for their readability, correctness, completeness, executable, and conformation to any appropriate design style standards. This validation ensures that the data can be readily used in the development process and helps prevent any unnecessary project or schedule delays due to incorrect or incomplete data. As was explained in the previous section for critical blocks, it is necessary to run the actual tools that will be used in the design to verify the assumptions and claims for block as well as chip level data. One important addition is to run code

coverage checks on the simulation testbenches to grade the testbench for completeness.

5. Design Feasibility Assessment

Design feasibility assessment is the analysis of a proposed BDD to determine the risks in accepting the design. The assessment process consists of two types of design analysis, the block assessment, performed on each block and the chip assessment done once for the whole chip.

5.1 Block Level Assessment

The block assessment predicts the expected values and error margins for each block used in the specified BDD. Design blocks can exist in various levels of resolution with respect to the final design (i.e., hard block, soft block, or firm block). Additionally, the design team's knowledge of and experience with a particular type of block can vary. Thus, the previous experience and the understanding of a block can contribute to a large range of error-bounds in the prediction of area, power, performance and cost across the blocks that will work together to form the final design. Hence the block level assessment is graded into three categories namely, coarse, medium and fine grain assessment. For example, computing the area of a block based on equations and parameters of nand equivalents, number of combinations gates and number of flip flops might be considered coarse grain, while, a quick synthesis of a block to get an assessment of area might be considered medium grain, while creating a hardened version of a soft block might be considered a fine grain assessment of the area of that block. The next few sections provide details on coarse grain estimates for area, power and timing.

5.1.1 Estimating Critical Design Parameters

In order to make accurate prediction of the design, several estimations need to be made for the critical design parameters. We will present some formulations that can be used to estimate these critical design parameters. Before performing any estimation, we need to extract some values from each block:

- Input data rate (*IDR*) – the rate of the data stream coming into the block.
- Activity rate (*ACT*) – the percentage of the logic that is active based on the function of the block.
- Number of flip-flops (*FF*)
- Number of combinational logic gates (*CG*)

5.1.2 Block Area Estimation

The area of soft and firm blocks can be estimated to the first order using the following formula.

$$\text{Block area} = \left\{ \sqrt{\left[\frac{CG \times G_{in} + FF \times G_{ff}}{K_{rf}} \right]} + 2 \times W_{pg} \right\}^2$$

Where, G_{in} is the grid size of 2 input NAND gates at the appropriate process, G_{ff} is the grid size of flip-flops, K_{rf} is the routing factor (a value ranging from 0.75 to 0.95 depending on numerous variables, such as the number of metal layers used for interconnection, existence of any hard macro inside the block, porosity of the standard cells used, design rules for vias, block size, block type, power requirements, performance requirements, and clock tree structure), and W_{pg} is the width of power and ground ring.

5.1.3 Block Power Estimation

There are three types of powers that need to be estimated: (i) the average switching power consumption due to the clock, (ii) the average switching power consumption due to logic signals, and (iii) the average switching power consumption due to I/O. Each prediction is described as follows:

$$\text{Ave. switching power consumption for clock} = (CTL \times C_i + FF \times C_c) \times K_c \times CKF \times Vdd^2$$

Where C_i is the capacitance of the input pin of the clock buffer cell, C_c is the capacitance of the clock pin of a flip-flop cell, K_c is the constant factor for clock tree routing layer parasitic capacitance, CKF is the clock frequency, and Vdd is the supply voltage.

$$\text{Ave. switching power consumption for logic gates} = 0.5 \times CG \times C_n \times K_s \times IDR \times ACT \times Vdd^2$$

Where C_n is the capacitance of the input pin of a two input NAND and K_s is the constant factor for signal routing layer parasitic capacitance.

The I/O power is a summation of the input pad power, the output pad power and the bi-directional pad power. Each power type is estimated as follows:

$$\text{Input pad power} = N_i \times ACT \times IDR \times 0.5 \times (C_{in} + C_{out} + C_{wire}) \times Vdd^2$$

Where N_i is the number of input pads, C_{in} is the input capacitance of the first logic encountered, C_{out} is the output capacitance of the pad, C_{wire} is the capacitance of the wire connecting the pad to the logic, and Vdd is the pad voltage.

$$\text{Output pad power} = N_o \times ACT \times ODR \times 0.5 \times (C_{out} + C_{load} + C_{wire}) \times Vdd^2$$

Where N_o is the number of output pads, ODR is the average output data rate (expressed as a frequency), C_{load} is the loading on the pad, and C_{wire} is the capacitance from the last logic gate to the output pad.

$$\text{Bi-directional pad power} = N_b \times ACT \times \{0.25 \times (C_{out} + C_{load} + C_{wire}) \times Vdd^2 \times ODR\} + \{0.25 \times (C_{in} + C_{out} + C_{wire}) \times Vdd^2 \times IDR\}$$

Where N_b is the number of bi-directional I/O pads and the rest of the parameter are similar to input pad power and output pad power

The average switching power consumption for the block is the sum of the average switching power consumption for the clock, logic gates, and I/O.

5.1.4 Block Timing Estimation

The setup/hold and delay times for I/O signals are determined by the logic depth between the boundary signals and their closest, connected register elements (flip-flops or latches), as well as the logic depth between the clock signal at the boundary and the closest, connected register elements. The logic level count (logic gate count) can be used to determine the setup/hold and delay times. The clock frequency (CKF) is determined by the specification that is defined for each block.

5.2 Chip Level Assessment

The chip assessment predicts expected values and error margins for a BBD, based on chip and block-level critical design parameters identified in customer specifications. These parameters include power, performance, area, cost, schedule and quality. Chip analysis helps the design team determine whether to accept or to reject a proposed BBD based on the calculated design risk.

The assessment is performed at three levels, namely, Initial, Refined and Final. At the initial assessment stage, information at the chip level is determined by a simple summation of information obtained for each block. As we have stated earlier, the information obtained from each block can be coarse, medium or fine grain depending on the quality of information available for each block.

At the refined assessment stage, a weighting factor is assigned to each of the block in order to assess more accurately, the contribution that this block might make to the overall chip parameter. Blocks might be broken up into three buckets of small, medium and large for the

current mix of blocks based on their initial estimates. While it may be okay to have coarse grain estimates for small blocks, medium and large sized blocks might be evaluated with their medium or fine grain estimates as their contribution margins are significantly higher for the overall chip. Error margins on large or medium blocks may also contribute to greater standard deviations. Statistical and mathematically significant techniques are used to compute refined estimates.

In the final assessment phase, a complete pass is redone taking into consideration, impact from other critical design factors. Initial place and route data for critical blocks may also be done based on the criticality of the design parameters in the overall specifications. The final cost estimate for the project is then based on the data generated and is more tied to the risks identified in the project.

Cost estimation includes non-recurring engineering (NRE) cost and production per part (PPP) cost. NRE cost is the time and material cost to develop a chip, while PPP cost includes production mask set cost, die cost, package cost, assembly cost, test cost, IP royalty cost, and costs for shipping, banking and drying trays.

6. Project Planning and Design Budgeting

The results of the validation and assessment processes in FEA can be used to develop project plans, design budgets, and ultimately the project sign-off agreement with the customer. Within the context of BBD, project management usually focuses on the quick completion (for faster time to market) of a complex design. Based on the complexity and the risk level of each block, the project manager can plan (i) work distribution that matches designer experiences with the type of block, (ii) project dependency (i.e., parallel and sequential relationships between the blocks), (iii) project schedule that includes target deadline for each component, (iv) distribution of the resource among all designers, (v) design documentation and (vi) testing schedule for each module as well as the entire chip. Since the overall chip and all required blocks included in this chip have been analyzed, the projected target for completion will be more accurate.

6.1 Project Plans

Usually, various types of planning control documents and instruments are needed to complete complex projects successfully. A project plan usually describes what should be done, and a project management plan usually describes how it will be done.

The project plan must include a Statement of Project Scope, a work breakdown structure, a precedence/dependency graph, a project schedule, a resource plan, a documentation plan, a test plan, material and equipment lists, design data management plan and a communication plan that identifies milestones and reviews.

The project management plan should cover the standards, policies, and procedures that must be adopted to execute the project. It should also document the goals and objectives of the project. It must clearly indicate the actions that are required at each milestone, be explicit about tradeoffs choices and how they should be made, team organization, travel plans, risk abatement plans, identify the metrics that will be tracked in the project and what recourse will be taken in order to recover from failure points. Change management procedures should also be clearly articulated, as also how the project plan itself will be kept up to date.

6.2 Initial Design Budgets

Data created and refined during design feasibility analysis serve as the starting point for initial design budgets. Sufficient margins must be maintained to ensure closure. Budgets are developed for critical design parameters including cost, performance, timing, area, power, test, schedule and quality. The initial block development plan is also created for each block. This plan must include the DFT strategy for each block, the block collaring requirements, and the tests that must be merged. Preliminary estimates for expected test coverage and expected tester time, and expected vector set length are also included at this stage.

6.3 Customer Signoff

This step outlines the final step of front-end acceptance. The design team meets with the customer to agree on the next steps for the project provided both parties agree on the contract. Items that must be documented at this stage include, project scope and definition, work to be performed, acceptance criteria, customer responsibilities, design center responsibilities, block acquisition plan, Risk assessment and abatement plans, Assumptions, project plans, Intellectual Property rights, and the payment schedule.

7. Conclusions

In this paper, we have presented a front-end acceptance methodology that is used as the initial phase of the block-based design methodology. FEA is useful to analyze the design feasibility as well as estimating design risks directly affecting time to market. By performing comprehensive data validation and quality estimation, the project can be scheduled more accurately, and realistically. Although the FEA process adds to the development cycle and the overall time-to-market, the return on investment (ROI) for the time spent on FEA can be significant, due to the increased accuracy in the design estimates and a reduction in design risks and errors.

8. Acknowledgements

This methodology is part of a patented methodology called Block Based Design. The authors would like to thank and acknowledge the contributions of the other authors of the patented methodology.

9. References:

- [1] Defining Platform Based Design By Alberto Sangioavanni-Vincentelli, EE design, Feb 5th 2002. <http://www.eedesign.com/story/EG20020204S0062>
- [2] Developing and using a Field of Experience in BBD; A Cadence White Paper by C.Lennard and H.Chang