Low Power Scheduling for High-Level Synthesis

Muhammad Farid Izwan Bin Mohamad Shabri  
Hochschule Hamm-Lippstadt59557 Lippstadt, Germany  
[muhammad-farid-izwan.bin-mohamad-shabri@stud.hshl.de](mailto:muhammad-farid-izwan.bin-mohamad-shabri@stud.hshl.de)

*Abstract*—This paper explains how low power scheduling is used for High-Level Synthesis . Scheduling is one of many stages for high-level synthesis. By using low power scheduling, switching activity is reduced which meet the time constraints. More details about scheduling are written in this paper, so that we clearly understand the benefits of low power scheduling for High-Level Synthesis. (*Abstract*)

Diagram

Description automatically generatedKeywords—component, formatting, style, styling, insert (key words)

# Introduction (*Heading 1*)

In this world full of advanced technology, microelectronics is very vital and act as a backbone to the development of software and hardware systems in recent decades. All inventions and innovations that are made by humans in the past are inspirations and the starting point to the more advanced technological changes today that produce a complex system. A complex system that full of design layers and architecture need years of effort to make a system become real and not just a stupid ideology from an optimist or idealist. Challenges and failures that come by, become a great lesson to create a great and reliable system for a very long time. All the hardworks in creating model and prototype must be must be appreciated because without this effort, the resulting system would have been full of errors and less reliability.   
 The real embedded system, integrated circuit IC and robot automation are example of hot topics that many developers and engineers discuss on how to improve the creation of new advanced technology. Design, Fabrication, Testing and Packaging are four stages in the creation of integrated circuit.

# High level synthesis

## Definition

High-level Synthesis is a translation process from behavioral description into a structural description. Scheduling, allocation and binding are three phases of High-Level Synthesis. The phases are randomly ordered based on design flow and are used to support decision making on design stage , namely size, performance and for this topic, we focus on power consumption that has become more and more important in recent years. When talking about power consumptions, it means that we wanted to optimize the usage of energy for a system so that it become more efficient. The main purpose of low power scheduling in High-Level Synthesis is to schedule operations while minimizing switching activity and select low power modules while meeting the time constraints.

## Fundamentals of High-Level Synthesis

As mentioned, the three phases are scheduling, allocation and binding. In scheduling phase, the time for operation to be executed is determined. The allocation phase determines how many instances of each resources are required and the binding or also well-known as assignment phase selects on what resource a computational operation will be run. Figures 1 shows design phases of the High-Level Synthesis.

Most of the time, the input for High-Level Synthesis is a control data flow graph, called CDFG.

# Power consumption and dissipation

In recent years, the demand of personal computing devices and wireless communications equipment has increased. As a result, there is a greater demand for low-power circuit design. Along with area and run-time, lowering power consumption has become one of the most essential criteria. There are some reasons why the consideration of low power is very strong :

- Increased demand of portable systems which require the battery life to be more long lasting. Laptops and mobile phones are the big picture for this issue.

- Thermal considerations which means that the cost of cooling and packaging would be reduced if low power dissipation

-Environmental issues because when smaller power is dissipated, lower the heat pumped into the rooms which means that usage of electricity is reduced, hence give less impact on environment

The source of power consumption or dissipation in digital CMOS circuit are as shown in figure below:

Diagram

Description automatically generated.

The parameters that are independent and not related to each other to affect power consumption as well as energy usage are:

- supply voltage

-the clock frequency

-the switching activity per clock cycle at various signals in the circuit.

-the parasitic capacitance

Text

Description automatically generated with low confidenceThe parameters above are proven by an expression, called dynamic power dissipation expression.

CL is the load capacitor, Vdd is the supply voltage,αis the average or expected number of transitions per clock cycle also known as the switching activity, and f is the clock frequency

Hence, to optimize the power consumptions, the factors must be tackled. Here are the suggestions or ideas:

- using the smallest possible supply voltage.

- parallelism and pipelining mechanism to lower required frequency operation.

- power control by disconnecting the power source when the system is in idle mode.

# Scheduling algorithms in high-Level synthesis

Table

Description automatically generated with medium confidenceScheduling is one of the phases in High-Level Synthesis. Here are the classification ofscheduling as shown in figure below.

## Scheduling Algorithm with Multiple Supply Voltage

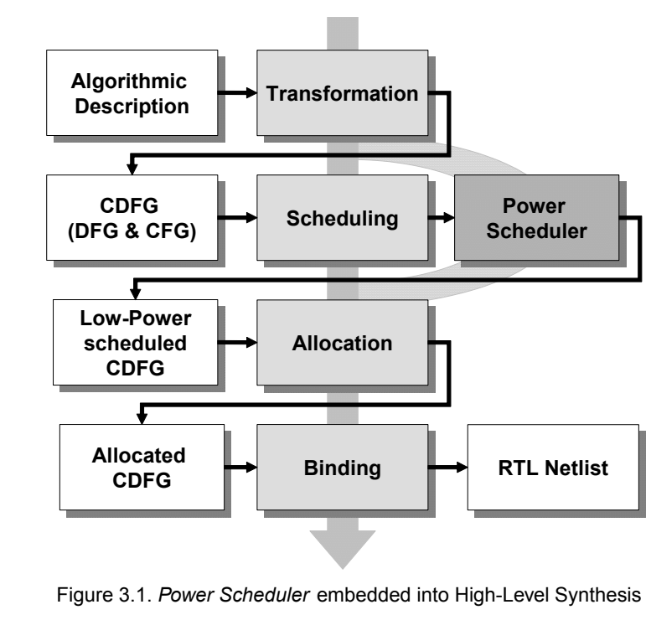
## Scheduling Algorithm based on Frequency Variation

## Scheduling Algorithm based on Switching Activity

# low power scheduling

The focus in this topic is how power consumption is optimized therefore low power is dissipated in scheduling task in High-Level Synthesis. The growing use of battery-powered and often wireless portable systems is boosting demand for low-power IC and SoC technologies. Furthermore, users have come to expect more capability and longer battery life from such systems. A modern cellular phone, for example, might have capabilities like the ability to operate as a personal digital assistant, storing data such as photos and connecting to the Internet. Implementing a stable power network and limiting power dissipation have become key issues for design engineers in today's exceedingly big and complicated architectures.

## Methodology analysis

To integrate low power methods in High-Level Synthesis, we developed a system called Power Scheduler, which shows in figure below.

As shown in figure above, we can see the effect or impact of Power Scheduler in scheduling process. It replaces the existing scheduler of High-Level Synthesis system which causes low power consumption at scheduling process before entering allocation phase.The scheduled Control Data Flow Graph is generated which support low power. To generate the graph by using partition method, so that each partition can be activated or deactivated, while combining partitions to be run, hence minimizing the additional control components which causes less energy usage.

## Power reducton method

* Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as “3.5-inch disk drive”.
* Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
* Do not mix complete spellings and abbreviations of units: “Wb/m2” or “webers per square meter”, not “webers/m2”. Spell out units when they appear in text: “. . . a few henries”, not “. . . a few H”.

Identify applicable funding agency here. If none, delete this text box.

subheads should be introduced. Styles named “Heading 1”, “Heading 2”, “Heading 3”, and “Heading 4” are prescribed.

## Design Example

#### Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation “Fig. 1”, even at the beginning of a sentence.

## Standard Scheduling

As discussed in section 3 the second step of the developedPower Scheduleris the calculation of the ASAP and ALAP schedules. Both schedules are notreally applied to the CDFG. There are only needed to calculate the mobilityof each node. This will discussed later in this section.ASAP means, all operations are scheduled as-soon-as-possible in the timeschedule. Figure 11 shows the ASAP schedule for the example design givenin section5. Respectively, ALAP means to schedule the operations of a DFGas-late-as-possible in time. The ALAP schedule of the design example isshown in figure 12. In both cases we make the following assumptions for theexample:•all operations in the DFG have a delay of 1 and•join or fork components need one timestep for their operation.After ASAP and ALAP scheduling it is possible to calculate the so calledmobilityfor each node in the DFG. This mobility measures the degree offreedom for a node to be scheduled. The mobility for a nodekof the DFG isdefinedasmobility(k) =ALAP(k)−ASAP(k),(3.10)whereasALAP(k)give the timestep where nodekis active by ALAP schedul-ing andASAP(k)give the timestep where nodekis active by ASAP schedul-ing. Nodes with mobility equal 0 are fixed, because they are scheduled at thesame timestep within ASAP and ALAP.For example, the addition operation behind the constant coefficient multipli-cationcm[0]is scheduled by ALAP at timestep 2 and by ASAP at timestep 1.The mobility of this operation is 1. That means, the node can be scheduled at one timestep.For the developed method it is necessary to calculate for all nodes of the DFGthe mobility. In section 10 of this chapter, it will be shown how important themobility is for thePower Scheduler. The idea is to schedule nodes to sametimestep within the DFG and to combine them to partitions.

## Path Determination

The third important step of thePower Scheduleris the determination of pathswithin the DFG. As described previously in section3 this step consist of threedifferent phases. The first phase contains the analysis of independent paths,see 8.1. In the second phase paths between fork and join nodes are examined,see section 8.2. And finally, the third phase examines the impact to schedulecontrol nodes for the DFG, see section 8.3.Before, we start with the detailed discussion of the paths determination, wehave to define what a path is for thePower Scheduler.

## Compatibility Graph

In this chapter a compatibility graph is described. The nodes of the graphconsist of all paths from the path determination of thePower Scheduler. Theedges reflect the compatibility between the paths. That means, if two or morepaths are compatible which each other they can be combined to a partition.But when paths are compatible? This depends on Path Time (see defini-tion 8.2) and on the mobility of the path resp. the nodes inside the path. Insection 7 we saw how the mobility for a node is calculated. For the defini-tion of the path mobility, we don’t take into consideration the mobility of theother paths. That means, by moving a node downward a subsequent nodemay get mobility 0 and this will influence the path mobility. However, thiswill not be recognized here, it will be done later by the construction of thecompatibility graph. If it will be done at this time the degree of freedom forfinding an optimal solution with our approach is reduced. Now we can definethe mobility for a path.

## Clique Problem

In this section we will briefly describe at first the clique search problem. Af-terwards, we will discuss strategies and solutions for the clique search. Oneof these methods is selected for thePower Scheduler. A clique consists ofdifferent paths that can be combined to a partition. The partition can be acti-vated or deactivated to reduce power consumption. As discussed before theidea is to build great partitions to reduce the control overhead for integrat-ing the power reduction methods. Furthermore, we have to integrate into theclique algorithm that all nodes are scheduled according equation 2. This weachieve by deletion all nodes that are in timing and dataflow conflict to thatnodes that are in a clique. That means, if we found a clique we will deletethose nodes that are in timing and dataflow conflict to the nodes in the clique.as

## Experimental Result

In the last session, we show how the DFG is partitioned with respect to re-duce the additional power cost. We found five partitions for our exampledesign by the proposed algorithm, see figure 33. Let us assume that anaddition operation has a power consumption ofPdynamic,add=20μW/MHzand a multiplication consumesPdynamic,mul=60μW/MHz. For the join ele-ments isPdynamic,join=10μW/MHz. Furthermore, let be the power reductionmethod we apply here to our design gated clock with a power consumptionofPgc,p=6μW/MHz. Table 3 shows the partitions and the energy. Hence, inour calculation model we take the worst case into account. That means, weassume to have a switching activity at each clock cycle, see also section 6.8.

# Summary & Conclusion

In this chapter, we presented an approach for low power driven synthesis.ThePower Schedulerreads a CDFG and writes a scheduled and partitionedCDFG. Besides the standard scheduling approaches, a path determination isapplied which is the basis for the design partitioning. Each partition allowsthe integration of dedicated turn-on and turn-off mechanisms into the design.That means, if a partition is not active, it can be turned off to reduce power consumption and therefore energy. With the proposed power methods we areable to reduce besides the dynamic the static power consumption by usingpower down.

The design example we are using is part of the MPEG-2 algorithm. All de-veloped methods are discussed with the example. The example shows theimpact and effectiveness of thePower Scheduler. Further examples, filteralgorithms emphasize this additionally.

##### References

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.