A New Approach for Low Power Scheduling in a VLIW Architecture Operating at Multiple Voltages

Chung Yung, Ming-Sian Lin, and Ching-Ho Cheng.
Department of Computer Science and Information Engineering
National Dong Hwa University
Hualien, Taiwan, R.O.C.

Abstract

Both total power consumption and peak power are the two critical issues in low power design. Dynamic voltage scaling has shown its high efficiency in low power scheduling in recent researches. And peak power also can be solve by integer linear programming. In this paper, we develop a new framework for then combination of both dynamic voltage scaling and peak power optimization in VLIW architecture. We represent integer linear programming model for low power problem with timing and resource constraints.

1 Introduction

Power design issue has been a critical and limiting factor in the design process of microprocessors. Higher clock rates and increased device count directly result in more dynamic power consumption. Since the system total power consumption is determined by the equation:

$$P = \alpha C V_{dd}^2 f$$

Where α is the probability of bit switching. C is the load capacitance. V_{dd} is the supply voltage and f is the frequency. High frequency and voltage cause high power consumption. However, the reduction of both voltage and frequency levels give reduced power consumption, but also increasing the circuit delays. The tradeoff between power and performance is an important point that how to adjust voltage and frequency levels without greatly affecting performance.

The peak power is the maximum power consumption of the IC at any instance during its execution[15]. High peak power results in the unexpected situation of supply voltage levels [14]. It causes high current flow which leads to the reduction of supply voltage levels at different part of the circuits, because of high IR drop in the power line. The reliability of circuit will be reduced by the high peak power problems. Hence, the extra bigger sinks and costlier heat dissipation mechanism is need to maintain the operating temperature of the ICs in its tolerance limits.

In recently years, a new important dynamic power-management technique, dynamic voltage scaling(DVS), has been developed to save energy.

The tradeoff between power-performance can be tuned according to the time intension of different applications. The goal will be to minimize power consumption without missing performance needs.

Theoretical studies [5] and simulations[27, 26, 3, 8, 22] on the potential of DVS techniques has shown that DVS technique is a good way to balance both power and performance. In [29], Zhang et al. present a compiler-directed technique that takes advantage of schedule slacks to optimize leakage and dynamic energy consumption. Mohanty, in [13, 14], describes new integer linear programming(ILP) models and algorithms for datapath scheduling that aim at minimizing peak power while maintaining performance.

Low power VLSI design can be achieved at various design levels[8]. In this study, we focus on power and peak power minimization at the behavior level. The VILW is considered as the hardware architecture. For example Ti c6x series have four different function units. The scheduling must also consider resource allocation. Our goal is to minimize both total power consumption and peak power at any instance.

Section 2 gives the overall framework of combination of DVS and peak power scheduling. And section 3 shows the DVS ILP formulation that reduces the total power consumption. Finally section 4 presents peak power ILP formulation that solves peak power problem in region. And finally gives the analysis result.

2 Low Power Framework

In [27], the authors extend an existing mixed-integer linear program formulation for the scheduling problem by accurately accounting for DVS energy switching overhead, by providing finer-grained control on settings and by considering multiple data categories in the optimization. The basic analytic element is single entrance and exit basic block which is call region. They use profiling technique to counts the entrances of each region. The power mode instruction is inserted on the edge between regions to reduce power consumption without violating timing constraint. But the peak power problem is not taking into concerns.

However, Shiue [19, 20] extends traditional ILP approach that minimizes peak power while satisfying timing constraint. Given data flow graph(DFG) and component library, their ILP model produces the optimal peak power solution. But, the total power is never reduced.

Both of the two approaches above have optimal result in total power and peak power respectively. The previous power optimization researches, [26, 3, 8, 22, 13, 14] using DVS consider scaling voltage on the fly on each resource. But the practical XScale [4] shows that power scaling is through mode set instruction. Dynamic frequency clocking that provides different voltage levels for each resource at the same time is still not realized. Dynamic adjust power levels through mode setting instruction is more feasible by now.

So that we taking the advantage of [27, 19, 20] to develop a new combination framework that minimized total power and peak power through mode set instructions.

The system process will require two main steps of power optimization. The first step will be

the optimization of total power consumption by DVS. The next step will require the timing constraint that create before to get the peak power optimization of each region. Here is the overall process.

1. Step1: Get the result of DVS ILP formulation.
2. Step2: Determine the power mode and execution time interval of each region.
3. Step3: Calculate the intra scheduling of regions by Peak Power ILP forumlation.

As shown in Fig.1, The DVS scheduling take the control flow graph(CFG) as input to scheduling each region to reach the lowest required power without violating timing requirement. Since executing time of each region required longest can be determined from the DVS scheduling, we can extend the traditional ILP formulation for peak power into regions version. The new peak power ILP formulation must take the running time power mode setting into concerns. For the given power mode and execution time interval, the peak power ILP formulation the solves peak power problem.

3 Total Power Optimization

The DVS ILP formulation is based on [27]. The degradation of voltage and frequency level can lead to power consumption reduction, and increase delay time. The goal of the DVS scheduling is to minimize total power and achieve timing constraint. Fig.2 shows that the mode-set instruction is putting on the entrance of each region. And a profiling based approach, Fig.3, is developed to obverse the program behavior. The regions entrances statistics will be used for solving power minimization problem. The problem is then become where to put those mode-set instruction to gain the optimal power consumption.

Before we give the ILP formulation, the transition cost for power mode setting must be defined.

For the power mode transition cost from V_u to V_v

$$S_E = (1 - f) \times c \mid V_u^2 - V_v^2 \mid$$

$$S_T = \frac{2 \times c}{I_{MAX}} \mid V_u^2 - V_v^2 \mid$$

Where S_E is the transition cost for energy and S_T for time. f is the energy efficiency of voltage regulator. c is the voltage regulator capacitance. I_{MAX} is the maximum allowed current.

For the notations that are used in DVS ILP, here are their definitions.

1. R = the number of regions, that is, nodes such as basic blocks in a control-flow graph. It is different for different programs.
2. M = the number of mode settings. It is determined by the processor's available mode settings.
3. k_{uvm} = the mode variable for mode m on edge(u, v). $k_{uvm} = 1$ if and only if the mode-set instruction along edge(u, v) sets the mode to m as a result of the scheduling, and is 0 otherwise. For example, if edge u, v chooses mode 2, the $k_{ij2} = 1$, and other $k_{uvm} = 0$ (m $\neq$ 2). $\overrightarrow{k_{uv}}$ is the set of mode variables (M in all) for edge(u, v).
4. W_{vm} = the energy consumption for a single invocation of region v under mode m.

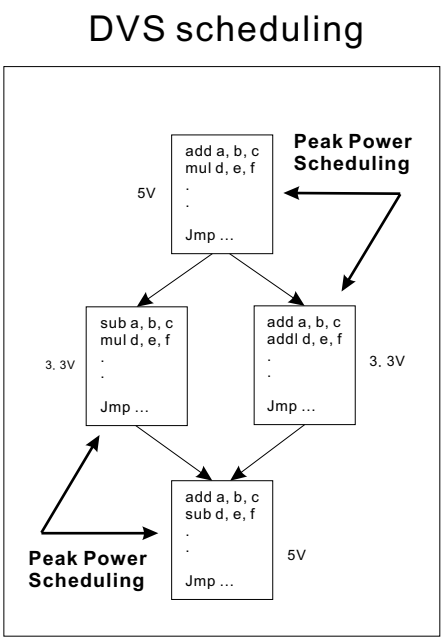

Figure 1: A combination Architecture of DVS and Peak Power Optimization

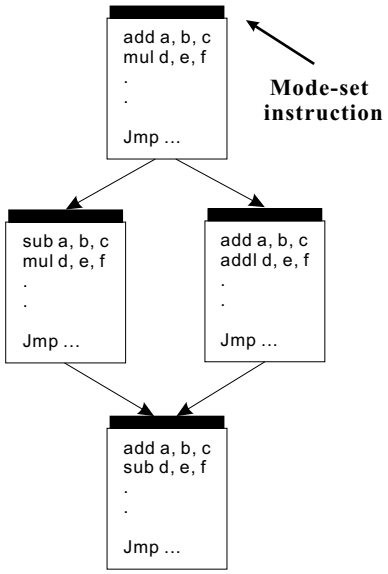

Figure 2: The Putting of Mode-set Instruction

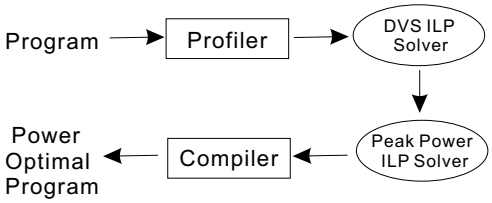

Figure 3: Program Profiling

5. G_{uv} = the number of times region v is entered through edge(u, v).

6. D_{wuv} = the number of times region u is entered through edge(w, v) and exited through edge(u, v).

7. T_{vm} = the execution time for a single invocation of region v under mode m.

If we let V_m be the supply voltage of mode m, then S_E is the transition energy cost for one mode transition, such that

$$S_E(\overrightarrow{k_{wu}}, \overrightarrow{k_{uv}}) =$$
$$c \times (1-f) \mid \sum_{m=1}^{N} k_{wum} V_m^2 - \sum_{m=1}^{N} k_{uvm} V_m^2 \mid$$

Likewise, S_T, the transition time cost for one transition, is represented as.

$$S_T(\overrightarrow{k_{wu}}, \overrightarrow{k_{uv}}) =$$
$$\frac{2 \times c}{I_{MAX}} \mid \sum_{m=1}^{N} k_{wum} V_m^2 - \sum_{m=1}^{N} k_{uvm} V_m^2 \mid$$

With these formulated power transition cost, the DVS optimization problem can be solved by the following ILP formulation.

1. Objective Function: Minimize the total power consumption

$$Minimize: \sum_{u=1}^{R} \sum_{v=1}^{R} \sum_{i=m}^{N} k_{uvm} G_{uv} W_{vm} +$$
$$\sum_{w=1}^{R} \sum_{u=1}^{R} \sum_{i=v}^{R} D_{wuv} S_E(\overrightarrow{k_{wu}}, \overrightarrow{k_{uv}})$$

2. Timing Constraint: Satisfy program deadline constraint

$$\sum_{u=1}^{R} \sum_{v=1}^{R} \sum_{i=m}^{N} k_{uvm} G_{uv} T_{vm} +$$
$$\sum_{w=1}^{R} \sum_{u=1}^{R} \sum_{i=v}^{R} D_{wuv} S_T(\overrightarrow{k_{wu}}, \overrightarrow{k_{uv}}) \leq deadline.$$

Since the execution plus total transition cost can not exceed timing constraint, the total power is minimized to lower voltage as low as possible.

4 Peak Power Optimization

Under the result of ILP formulation of DVS, the time interval and power mode of each region are calculated. So that we will apply on the time interval and power mode give by DVS formulation result to determine the peak power scheduling of each region. Here are the notations.

n_r = number of operations in the data flow graph of region r.

o_i = operation i, $1 \leq i \leq n_r$.

$o_i \rightarrow o_l$, o_i = an immediate predecessor of o_l.

FU_f = function unit of type f.

$o_i \in FU_f$ if o_i can be executed by FU_f.

A_f = integer variables that denote the maximum number of functional units of type f required in all steps.

$x_{i,j}$ = binary variable associated with o_i, where $x_{i,j} = 1$, if o_i is scheduled into step j, otherwise, $x_{i,j} = 0$.

S_i = starting step of o_i.

E_i = ending step of o_i.

P(i, m) = power consumption of operation i in power mode m.

D(i, m) = delay time of operation i in power mode m.

TS_r = the start time of region r.

TE_r = the end time for a single invocation of region r.

C_{total} = total number of control steps.

The main ideal of peak power scheduling is to give dependence, timing, and resource constraints. And minimize each control step peak power by rescheduling those mobile operations. This action cannot reduce power consumption but peak power in an instance. Here is the ILP formulation for peak power. An iteration will be performed unit all regions is solved.

1. Objective Function : Minimize the peak power
   Minimize $PeakPower$

2. Node Constraint: Each operation can only be issued in some control and use one function unit in VLIW architecture once.
   $\sum_j x_{i,j} = 1$, for all operation i in region

3. Peak-Power Constraint: Summarize each control step power in the execution time interval of each region. Since our goal is minimize peak power. The power consumption should not over than our objective function $PeakPower$.
   $\sum_i \sum_{S_i+D(i,m)-1 \geq j} \mathrm{P}(i, \quad m) \cdot x_{i,j} \quad \leq$ $PeakPower$, for $TS_v \leq j \leq TE_v$.

4. Resource-Constraint: Each VLIW function unit can only assign to one operation at a time. So that the total active function unit should not exceed total available resource.
   $\sum_{o_i \in FU_f} \sum_{S_i+D(i,m)-1 \geq j} x_{i,j} \leq A_f$,
   for all type f, and $TS_v \leq j \leq TE_v$.

5. Dependency Constraint: The program semantic should be preserved. So that our scheduling must keep the data dependency between operations. If there are data flow from o_l to o_i then o_i should not be executed before o_l is finished.
   $E_i < S_l$, for $o_i \rightarrow o_l$,
   where $E_i = \sum_j (j + D(i, m) - 1) \cdot x_{i,j}$ and
   $S_l = \sum_j j \cdot x_{i,j}$

6. Time Constraint: Those operations without successor should finish its execution before the end time of this region.
   $E_i \leq C_{total}$, for all o_i without successor.

5 Analysis

Since N input for the peak power instruction scheduling will produce N^N computation result in time complexity. If the average size of regions is $\frac{N}{k}$ for k regions. Then the peak power scheduling problem will be reduce to k $\times$ $\frac{N}{k}^{\frac{N}{k}}$. Hence, the polynomial degree will be reduced from N to $\frac{N}{K}$. For real world application, although we cannot give the exponential time solution, but a reduced polynomial degree version is given.

Compare to those peak power optimization in [13, 14, 19, 20]. A global version ILP formulation is given. They consider fully program as the basic element that take data flow graph as input. And output the minimized peak power scheduling base on the fully program execution. However, the key point for the peak power scheduling is to avoid power consumptive operation in the same control step with dependency kept. Since not only data dependency should be kept but

also control dependency. Their scheduling approaches will cause control dependence disappearance. This also causes wrong program behavior.

The second, with control dependency is kept, the mobility of each operation will only in the basic block. So that a global version scheduling cannot minimize peak power in some regions if the other utilize fully peak power. This will cause the global peak power scheduling allows peak power of regions as much as calculated peak power. In the contrast, a local peak power scheduling will be more efficient in reducing peak power in regions. Because it only considers local optimization, peak power of one region cannot affect another. It keep peak power of each regions as less as possible.

6 Conclusion

In this paper, we give a combination framework to solve both total power and peak power problem. DVS ILP formulation and peak power ILP formulation are integrated together. The DVS scheduling utilize profiling based technique to observe program behavior to insert power mode-set instruction. It reduce total power consumption with violate given timing constraint. The peak power scheduling that rearranges operation in regions to gain an optimal peak power consumption in an instance. The analysis shows that the combination framework is more flexibility in practices.

References

[1] Alidina, M., Monteiro, J., Devadas, S., Ghosh, A., Papaefthymion, M.: Precomputation-Based Sequential Logic Optimization for Low Power, *IEEE Transactions on Very Large Scale Integration Systems*, **2**(4):426-436, December 1994.

[2] Chang, J.M., Pedram, M.: Register Allocation and Binding for Low Power, *Proceedings of Design Automation Conference*, pp. 29-35, June 1995.

[3] Hong, I., Kirovski, D., Qu, G., Potkonjak, M.: Power Optimization of Variable Voltage Coro-Based Systems, *Proceedings of the 35th Annual Conference on Design Automation*, pp. 176-181, May 1998.

[4] Intel Corp. Intel Xscale Core Developer's Manual.: Available at http:://developer.intel.com/design/intelxscale, 2003

[5] Ishihara, T., Yasuura, H.: Voltage Scheduling problem for dynamically variable voltage processor, *ISLPED*, pp. 197-202, August 1998.

[6] Lee, C., Lee, J. K., Hwang, T., Tsai S-C.: Compiler Optimization on VLIW Instruction Scheduling for Low Power, *ACM Transactions on Design Automation of Electronic Systems*, **8**(2):252-268, April 2003.

[7] Lee, M. T-C., Tiwari, V., Malik, S., Fujita, M.: Power Analysis and Minimization Techniques for Embedded DSP Software, *IEEE Transaction on VLSI Systems*, **5**(1):123-133, March 1997.

[8] Lin, Y.R., Hwang, C-T., Wu, A.C-H.: Scheduling Techniques for Variable Voltage Low Power Designs, *ACM Transactions on Design Automation of Electronic Systems*, **2**(2):81-97, April 1997.

[9] Martin, R.S., Knight, J.P.: PASSOS: A Different Approach for Assignment and Scheduling for Power, Area and Speed Optimization in High-Level Synthesis, *Proceedings of the 37th Midwest Symposium on Circuits and Systems*, **1**:339-342, August 1994.

[10] Martin, R.S., Knight, J.P.: Using SPICE and Behavioral Synthesis Tools to Optimize ASIC's Peak Power Consumption. *Proceedings of the 38th Midwest Symposium on Circuits and Systems*, pp. 1209-1212, August 1995.

[11] Martin, R.S., Knight, J.P.: Optimizing Power in ASIC Behavioral Synthesis. *IEEE Design & Test of Computers*, 13(2):58-70, 1996.

[12] Mohanty, S.P., Ranganathan, N.: A Framework for Energy and Transient Power Reduction During Behavioral Synthesis, *Proceedings of International Conference on VLSI Design*, pp. 539-545, January 2003.

[13] Mohanty, S.P., Ranganathan, N.: Peak Power Minimization through Datapath Scheduling, *Proceedings of the 2003 IEEE CS Annual Symposium on VLSI*, pp. 121-126, February 2003.

[14] Mohanty, S.P., Ranganathan, N., Chappidi, S.K.: Low Power: Simultaneous Peak and Average Power Minimization During Datapath Scheduling for DSP Processors, *Proceedings of the 13th ACM Great Lakes symposium on VLSI* , pp. 215-220, April 2003.

[15] Mohanty, S.P., Ranganathan, N., Chappidi, S.K.: An ILP-Based Scheduling Scheme for Energy Efficient High Performance Datapath Synthesis, *Proceedings of the 2003 International Symposium on Circuits and Systems*, **5**:25-28, May 2003.

[16] Mohanty, S.P., Ranganathan, N., Chappidi, S.K.: ILP Models for Energy and Transient Power Minimization during Behavioral Synthesis, *Proceedings of the 17th IEEE International Conference on VLSI Design*, pp. 745-748, January 2004.

[17] Jain, V., Rele, S., Pande, S., Ramanujam, J.: Code Restructuring for Improving Execution Efficiency, Code Size and Power Consumption for Embedded DSPs. *12th International Workshop on Languages and Compilers for Parallel Computing*, 1999.

[18] Raghunathan, V., Ravi, S., Raghunathan, A., Lakshminarayana, G.: Transient Power Management through High Level Synthesis, *Proceedings of the 2001 International Conference on Computer Aided Design*, pp 545-552, November 2001.

[19] Shiue, W.T.: High Level Synthesis for Peak Power Minimization Using ILP. *Proceedings of IEEE International Conference on Application Specific Systems, Architectural and Processors*, pp. 103-112, June 2000.

[20] Shiue, W.T., Chakrabarti, C.: ILP Based Scheme for Low Power Scheduling and Resource Binding, *Proceedings of the 2000 IEEE International Symposium on Circuits and Systems*, **3**:279-282, May 2000.

[21] Shiue, W.T., Denison, J., Horak, A.: A Novel Scheduler for Low Power Scheduling and Resource Binding, *Proceeding of the 43rd Midwest Symposium on Circuits and Systems*, pp 312-315, August 2000.

[22] Shiue, W.T., Chakrabarti, C.: Low-Power Scheduling with Resources Operating at Multiple Voltages, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, **47**(6):536-543, June 2000.

[23] Sundararajan, V., Parhi, K.K.: Data Transmission over a Bus with Peak-Limited Transition Activity. *Proceedings of the 2000 Asia and South Pacific Design Automation Conference*, pp. 221-224, January 2000.

[24] Tiwari, V., Malik, S., Wolfe, A.: Instruction Level Power Analysis and Optimization of Software, *Journal of VLSI Signal Processing*, pp.1-18, 1996.

[25] Tomiyama, H., Ishihara, T., inoue, A., Yasuura, H.: Instruction Scheduling to Reduce Switch Acitivity of Off-Chip Busses for Low-Power Systems with Caches, *Proc. IEICE*, pp. 2621-2629, December 1998.

[26] Wang, L., Jiang, Y., Selvaraj, H.: Synthesis Scheme for Low Power Designs with Multiple Supply Voltages by Heuristic Algorithms. *Proceedings of the 2004 International Conference on Information Technology: Coding and Computing*, **2**:829-833, April 2004.

[27] Xie, F., Martonosi, M., Malik, S.: Intraprogram Dynamic Voltage Scaling: Bounding Opportunities with Analytic Modeling. *ACM Transactions on Architecture and Code Optimization* **1**(3):323-367, September 2004.

[28] Yang, H., Gao, G.R., Leung, C.: On Achieving Balanced Power Consumption in Software Pipelined Loops, *Proceedings of the International Conference on Compilers, Architectures and Synthesis for Embedded Systems* pp. 210-217, October 2002.

[29] Zhang, W., Vijaykrishnan, N., Kandemir, M., Irwin, M. J., Duarte, D. Tsai, Y-F.: Exploiting VLIW Schedule Slack for Dynamic and Leakage Energy Reduction, *Procedings of the 34th annual ACM/IEEE international symposium on Microarchitecture*, pp. 102-113, 2001

[30] C. Yung, M. Lin, "Low Power Instruction Scheduling with a Pee-hole Peak Power Limit", *submit to ASIAN Symposium on Programming Languages and Systems*, Taipei, Taiwan, November 4-6, 2004.

[31] C. Yung, L. Wong, "Low Power Optimization for a Pipelined VLIW Architecture", 2005.