An Operation Rearrangement Technique for Power Optimization in VLIW Instruction Fetch*

Dongkun Shin, Jihong Kim and Naehyuck Chang

School of Computer Science and Engineering
Seoul National University
{sdk,jihong}@davinci.snu.ac.kr, naehyuck@snu.ac.kr

Abstract

In VLIW machines where a single instruction contains multiple operations, the power consumption during instruction fetches varies significantly depending on how the operations are arranged within the instruction. In this paper, we describe a post-pass operation rearrangement method that reduces the power consumption from the instruction-fetch datapath. The proposed method modifies operation placement orders within VLIW instructions so that the switching activity between successive instruction fetches is minimized. Our experiment shows that the switching activity can be reduced by 34% on average for benchmark programs.

We propose a post-pass optimization technique that can reduce switching activity at instruction bus by modifying operation placement orders within VLIW instructions so that the power consumption at the instruction-fetch datapath is minimized. The proposed method takes advantage of a VLIW machine's instruction encoding characteristic: VLIW CPUs can place the same operation in multiple operation slots within the VLIW instruction. We reorder given VLIW instructions to equivalent instructions that have less switching activity.

In VLIW processors, an instruction scheduling technique must weigh the switching activity from the on-chip instruction bus as well as the off-chip instruction bus. Since the width of the on-chip instruction bus is generally much larger than that of the off-chip instruction bus, if an instruction schedule were produced considering only the bit changes from the off-chip bus, it might be a bad schedule for the on-chip instruction bus.

The total number of bit changes SW^B from a basic block B during the instruction fetch phase at the instruction bus is given by the sum of two terms, SW^B_{cache} and SW^B_{mem}. SW^B_{cache} represents the number of bit changes at the internal instruction bus and SW_{mem} indicates the number of bit changes at the external instruction bus. Assuming the load capacitance ratio of the external instruction bus to the internal instruction bus is α, SW^B, in the number of bit transitions at the internal bus, is computed as $SW^B = SW^B_{cache} + \alpha \cdot SW^B_{mem}$. If operations are reordered, SW^B_{cache} and SW^B_{mem} are changed.

When we denote the set of basic blocks which can be built by rearranging the operations in a basic block B as $EQ(B)$, given a basic block B, the problem is to find an equivalent basic block B' such that $SW^{B'} \leq SW^{B''}$ for all $B'' \in EQ(B)$. We compute an optimal solution for this problem by converting the problem to the shortest path problem.

We have performed experiments using a VLIW digital signal processor, TMS320C6201 [1], from Texas Instruments. We have compared the average number of bit transitions per instruction fetch (BT/IF) between TI compiler generated programs (the default column in Table 1), and rearranged programs by the proposed operation rearrangement technique (the ORT column in Table 1). The number of bit transitions during the instruction fetch phase is reduced on an average by 34.3%.

Benchmark Program	BT/IF		
	default	ORT	Reduction
vector multiply	68.6	43.7	36.3%
FIR8	86.8	56.7	34.6%
IIR	71.7	51.7	28.0%
lattice analysis	88.4	58.2	34.2%
W_vec	89.5	57.1	36.3%
minerror	50.6	31.3	38.1%
biquad	78.1	52.3	33.0%
Average	76.9	50.6	34.3%

Table 1. Experimental results

Reference

[1] Texas Instruments. *TMS320C62xx CPU and Instruction Set*, 1997.

*This work was supported by Korea Research Foundation Grant (KRF-2000-041-E00287).