

An Operation Rearrangement Technique for Low-Power VLIW Instruction Fetch

Dongkun Shin* and Jihong Kim

**Computer Architecture Lab
School of Computer Science and Engineering
Seoul National University, Korea**

Outline

- **Motivations**
- **VLIW Instruction Encodings**
- **LOR Problem and Solution**
- **GOR Problem and Solution**
- **Experiment**
- **Conclusions**

Motivations

Many mobile devices are designed using **VLIW** processors for high performance, which usually consume more power than single-issue processors.

In digital CMOS circuits, **switching activity** accounts for over 90% of total power consumption.

We propose a post-pass optimization technique that can reduce switching activity during the instruction fetch phase in VLIW processors

VLIW Instruction Encoding-Uncompressed

Program

```

IADD /*IntU*/
|| FADD /*FpU*/
|| LOAD /*MemU*/
|| STORE /*MEMU*/

ISUB /*IntU*/
|| IMUL /*IntU*/

IADD /*IntU*/
|| BEG /*BrU*/
    
```

Alternative encoding

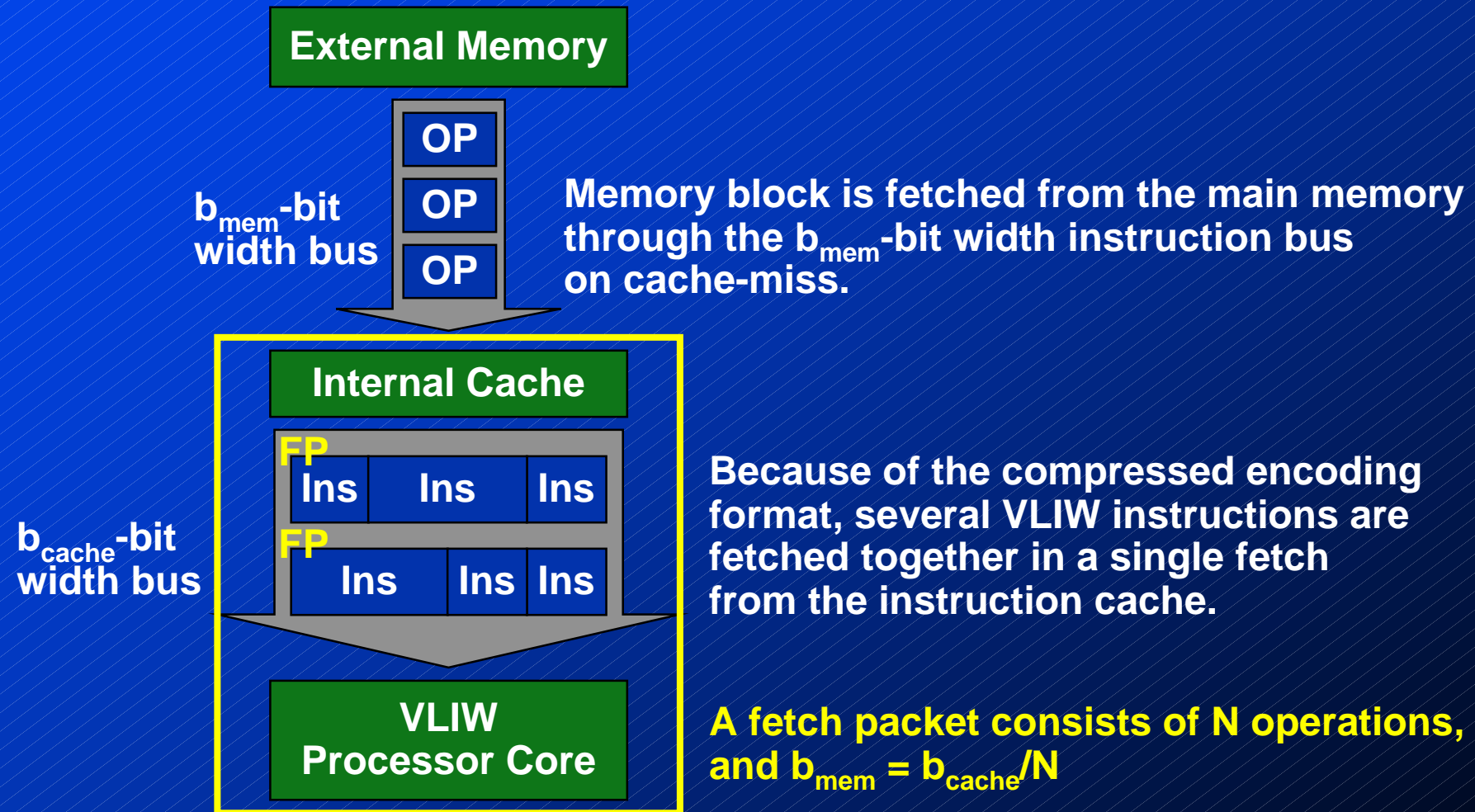
<i>IntU</i>	<i>IntU</i>	<i>FpU</i>	<i>FpU</i>	<i>MemU</i>	<i>MemU</i>	<i>CmpU</i>	<i>BrU</i>
-------------	-------------	------------	------------	-------------	-------------	-------------	------------

Functional Unit

IADD	NOP	FADD	NOP	LOAD	STORE	NOP	NOP
ISUB	IMUL	NOP	NOP	NOP	NOP	NOP	NOP
IADD	NOP	NOP	NOP	NOP	NOP	NOP	BEG

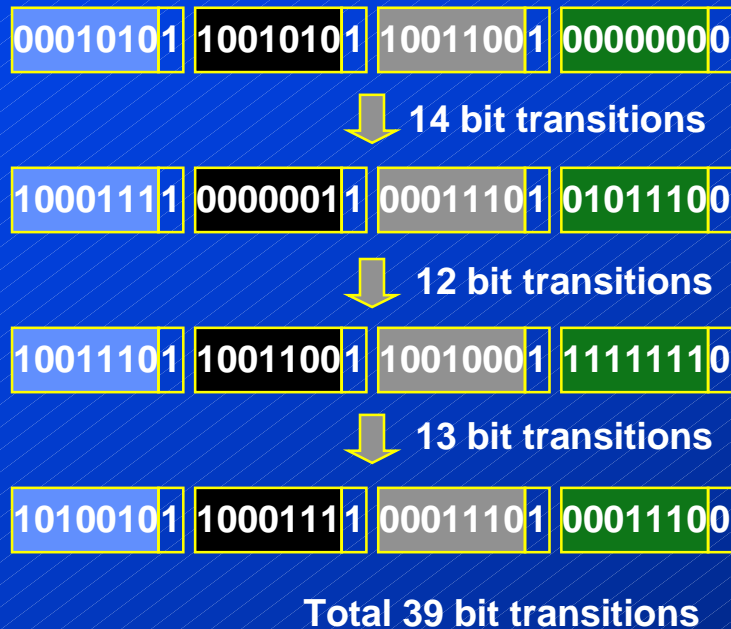
NOP	IADD	NOP	FADD	STORE	LOAD	NOP	NOP
IMUL	ISUB	NOP	NOP	NOP	NOP	NOP	NOP
IADD	NOP	NOP	NOP	NOP	NOP	BEG	NOP

Machine Model



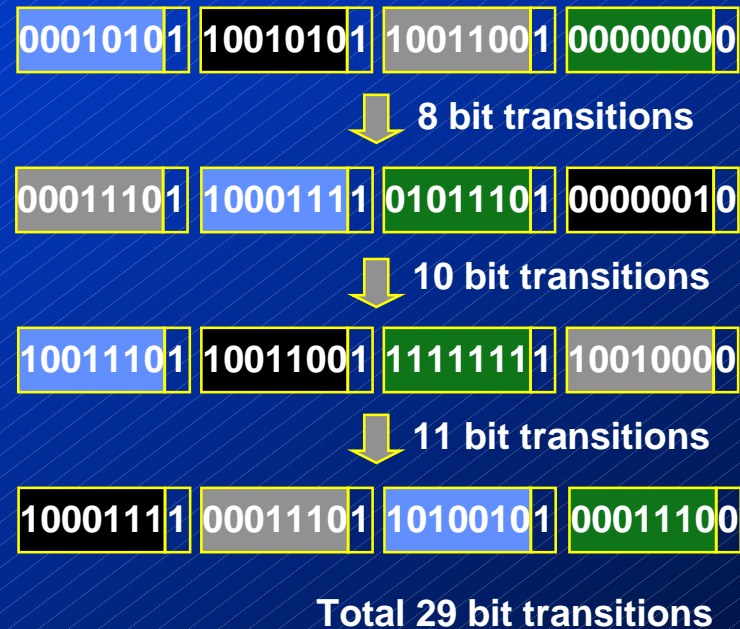
Basic Idea

Instruction Cache



(a) Before operation rearrangement

Instruction Cache



(b) After operation rearrangement

The total # of bit changes are reduced by 25%

Problem Formulation

Problem

how to reorder given VLIW instructions to reduce the number of bit transitions between successive instruction fetches.

Solutions

Local Operation Rearrangement (LOR) :
each basic block is independently considered.
Global Operation Rearrangement (GOR) :
all the basic blocks are simultaneously considered.

LOR Problem

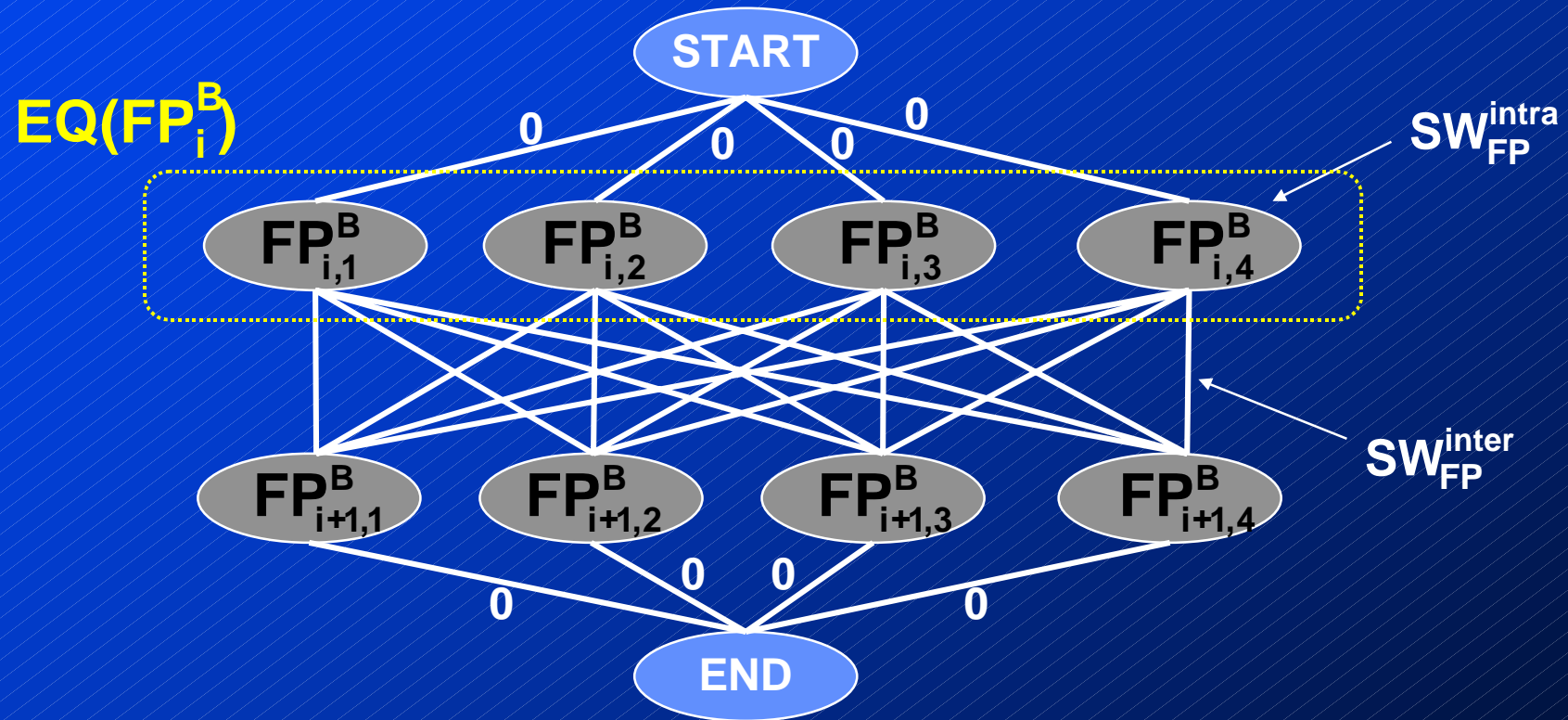
$$SW^B = SW_{cache}^B + \alpha \cdot SW_{mem}^B$$

α is the load capacitance ratio of the external instruction bus to the internal instruction bus.

SW_{cache}^B is the number of bit changes at the **internal** instruction bus.

SW_{mem}^B is the number of bit changes at the **external** instruction bus.

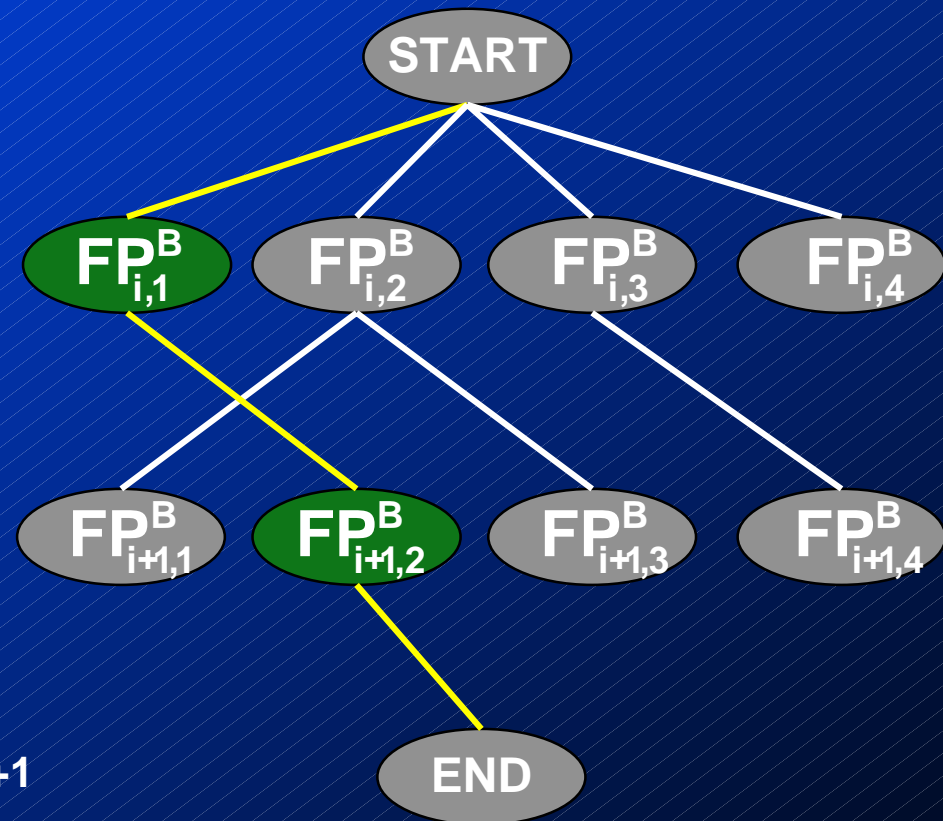
Solution for LOR



$EQ(FP_i^B)$: The set of equivalent fetch packets of FP_i^B .

Solution for LOR

- We find the **shortest path** from START to END, which is the solution of operation rearrangement to minimize the SW^B
- A node v_{i+1} in graph finds the node v_i through which the shortest path from START to the node v_{i+1} should pass.



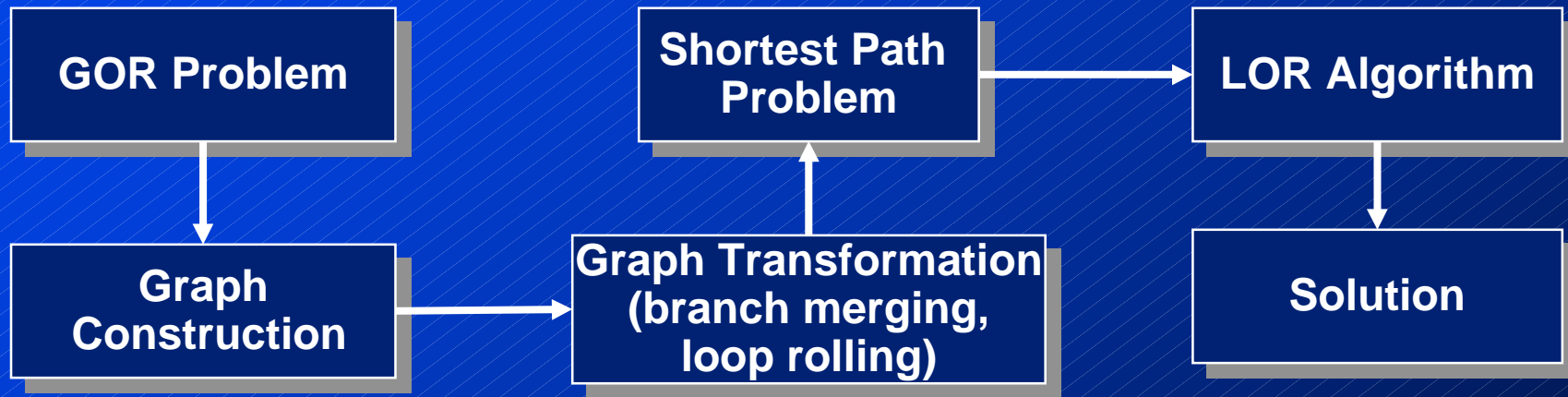
GOR Problem

- All the basic blocks in a program are simultaneously considered
 - how many times each basic block is executed.
 - how often each basic block experiences cache misses.
 - how basic blocks are related each other.

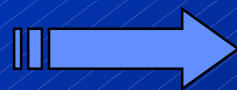
$$SW^S = \sum \sum SW_{BB}^{inter}(bb_i, bb_j) + \sum SW_{BB}^{intra}(bb_i)$$

- SW_{BB}^{inter} and SW_{BB}^{intra} is represented by SW_{FP}^{inter} , SW_{FP}^{intra} , weight of each basic block, and cache miss rate.

Solution for GOR



This method may require an excessive amount of memory and cycles.



We need a heuristic solution.

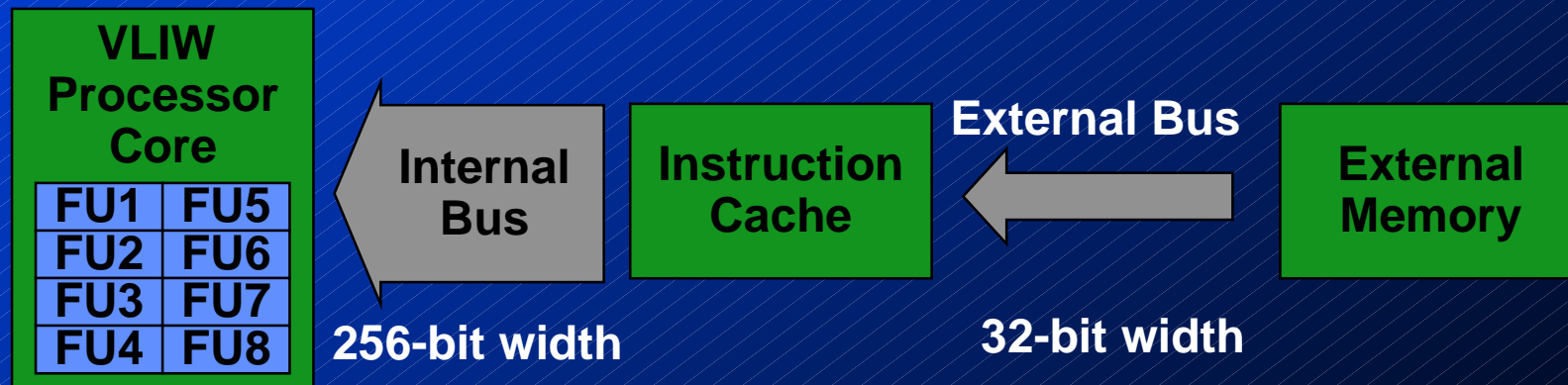
Heuristic for GOR

- All the basic blocks are not equally treated.
 - Basic blocks with larger effects on the total switching activity are more thoroughly reordered than ones with smaller effects.
- Not all the equivalent basic blocks in $EQ(bb_i)$ are tried to find an optimal solution.
 - Only N_{cand} equivalent basic blocks are created and included in graph.

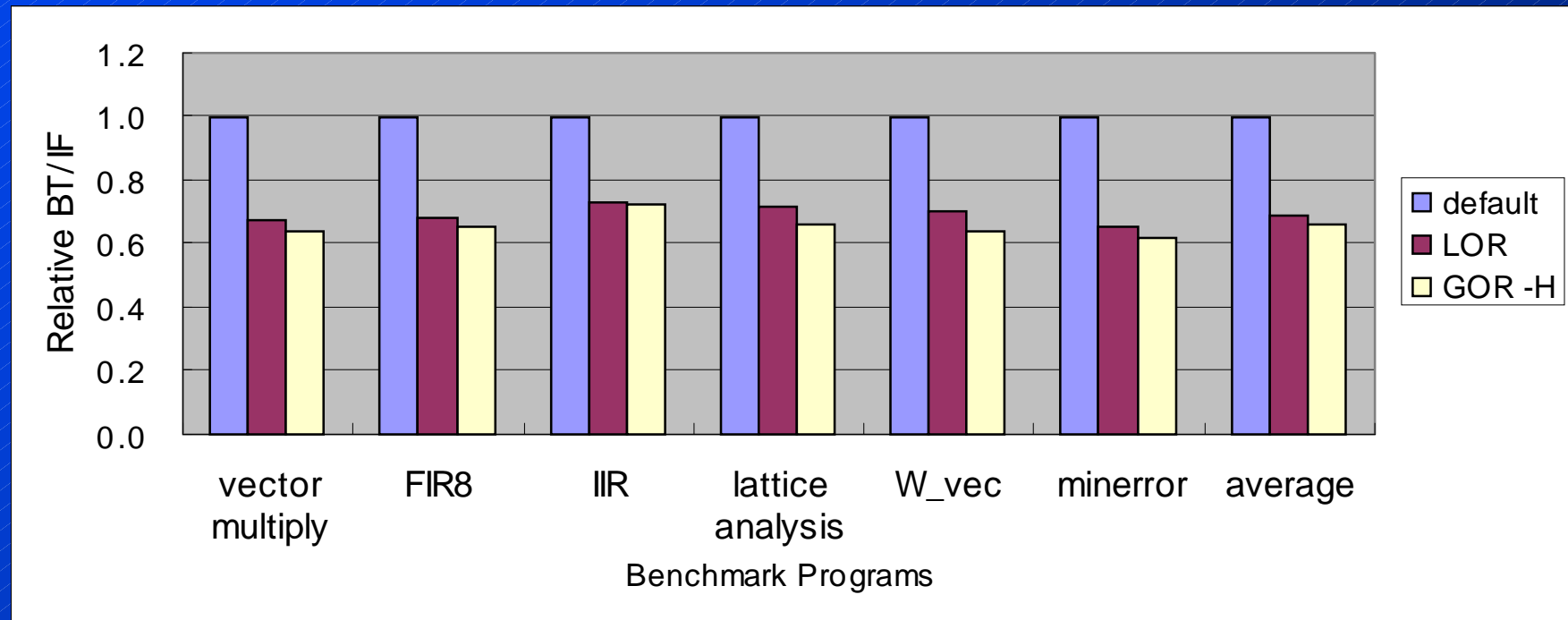
Experiment

TMS320C6201

- Fixed-point DSP
- VLIW processor that can specify **eight** 32-bit operations in a single 256-bit instruction.
- Use a compressed encoding



Experiment Results



For our benchmark programs, the bit transitions was reduced by 34% on an average.

Conclusions

- **Described a post-pass optimal operation rearrangement method for low-power VLIW instruction fetch.**
 - The switching activity was reduced by 34% on an average.
- **Future works**
 - The phase-ordering problem between the operation rearrangement and other compiler optimization steps.
 - Operation rearrangement problem in super-scalar processors.