

# Project Proposal

## **Topic:**

Instruction Encoding for Area minimalization of Instruction ROM

## **Description of Topic:**

Instruction encoding techniques for embedded system design, which encode immediate fields of instructions to reduce the size of an instruction memory. However, we have to add additional decoder for encoded values.

We can reduce memory size by two methods, first decreasing the number of words which depends on programmer and compiler.

The second methods is reducing the width of words which depends on system architecture. In an instruction word, an immediate field usually determines the width (or length) of instruction words.

However, there is no existing application that all values which can be represented by the immediate field are used. In this project, we try to implement instruction encoding techniques, which encode immediate fields of instructions to reduce the size of the instruction memory.

## **Why the Topic is important and interesting?**

Small memory size for instruction encoding means better use chip size. Smaller chips have much more advantage than bigger chips.

- They are small hence less material is required and cost price is reduced.
- They consume very less power.
- They can be cooled easily because they work on less power. One may argue according to stefan-boltzman law that surface area ( $A$ )  $\propto$  heat transfer per unit time ( $q$ ).

But in case of chips heat generated by them is bigger factor than heat transfer rate.

This topic also interesting because as we know moore's law is ending, we can no longer make chips small by doubling number of transistors on a microchip doubles about every two years. To achieve more smaller chips we have to find new and innovative solutions.

## **Description of methods used for evaluating the proposed idea:**

Original research applied encoding techniques to three embedded applications, ghostscript, mpeg2 decoder, and mpeg2 encoder. A code size of these application programs are about 42,000, 6,500, and 6,800 lines, respectively. The compiler is gcc-dlx 2 which based GNU CC Ver. 2.7.2 for DLX architecture.

We will use mips simulator for any implementation.

## **References to papers :**

[1] T.Okuma, H.Tomiyama, A.Inoue, E.Fajar, and H.Yasuura “Instruction Encoding Techniques for Area Minimization of Instruction ROM”,1998. (main paper)

[2] A. Grasselli. “The design of program-modifiable microprogrammed control units”. IRE Trans. on EC, EC-11:336339, 1962.

[3] J. L. Hennessy and D. A. Patterson. Computer Architecture: A Quantitative Approach. Morgan Kaufmann Publishers, Inc., 2nd edition, 1996.

## **Group details:**

Groups no: 4006

Members:

Name	Enrollment no	E-mail ID
Ankit Aharwal	18114006	aaharwal@cs.iitr.ac.in
Anuj Gupta	18114008	agupta6@cs.iitr.ac.in
Mohit Kumar	18114049	mkumar2@cs.iitr.ac.in
Navjit singh	18116051	nsingh1@ec.iitr.ac.in
Nikhil choudhary	18116054	nchoudhary2@ec.iitr.ac.in
Yashaswi Jaiswal	18114083	yjaiswal@cs.iitr.ac.in
Abhishek Kumar Gupta	18116001	agupta7@ec.iitr.ac.in