



**Sharif University of Technology**  
**Department of Computer Engineering**

# **Embedded System Design**

**Code-size Efficiency**

**A. Ejlali**

# Why Code-size efficiency?

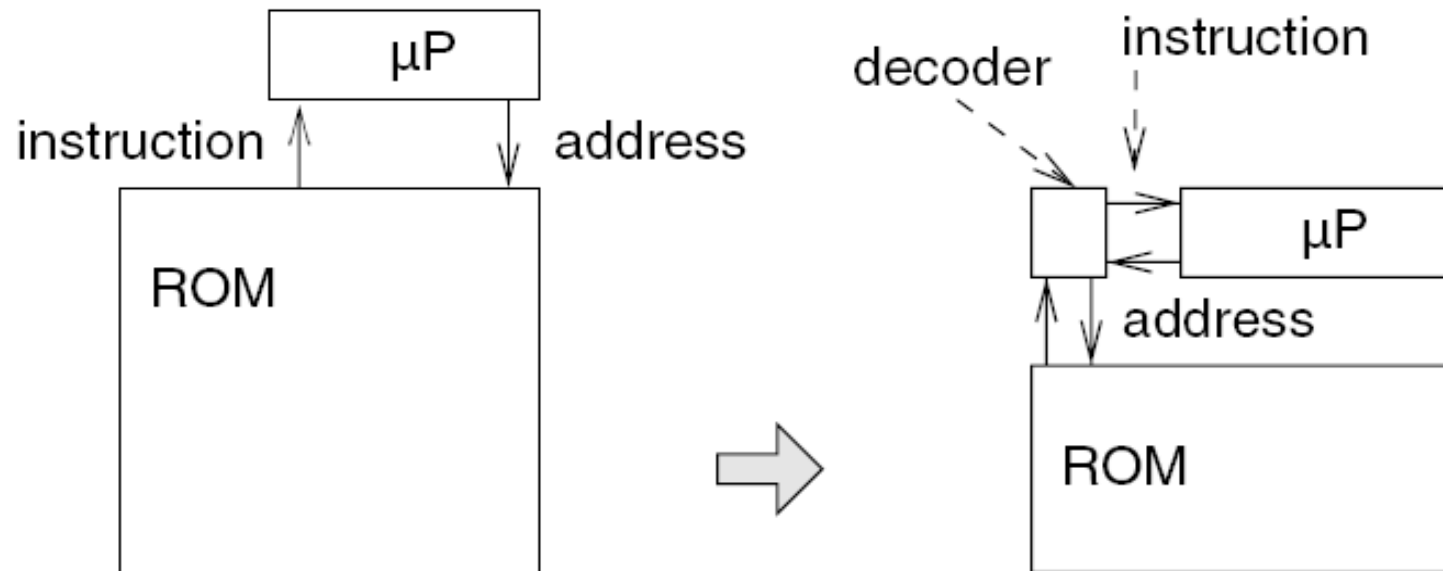
- **Minimizing the code size** is very important for ES, since:
  - HDDs are typically not available
  - Memory (ROM and RAM) and processor on the same chip (Embedded Memory)
    - e.g., SoC, Micro-Controllers

# Compression Techniques

- Reduces both the **area** and the **energy** necessary for fetching instructions.
- A **small and fast** decoder is used for the **instruction** memory to decompress the instructions **on the fly**.

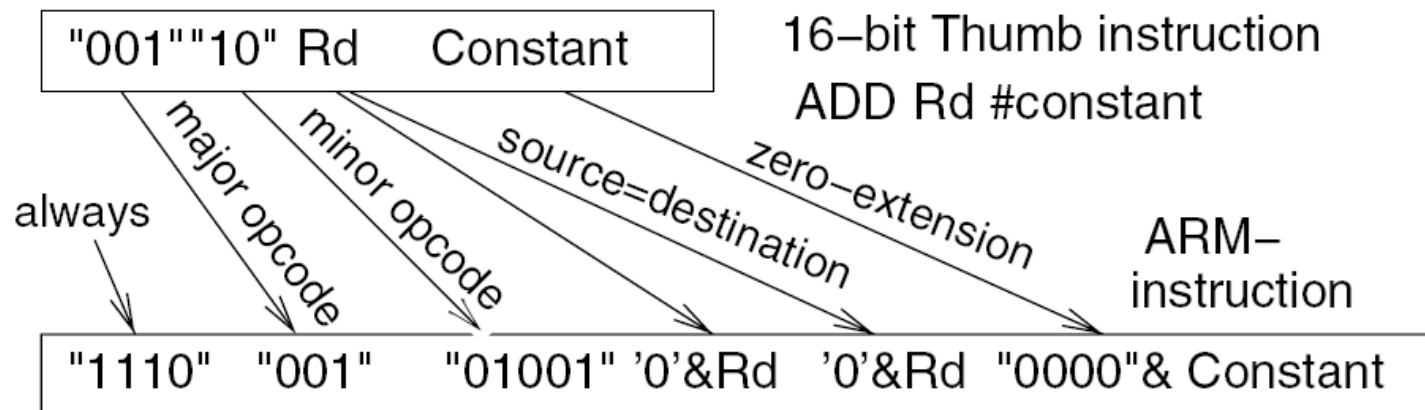
# Compression Techniques (Cont.)

- During decoding, **pipelining** can be used to keep the run-time penalty low.



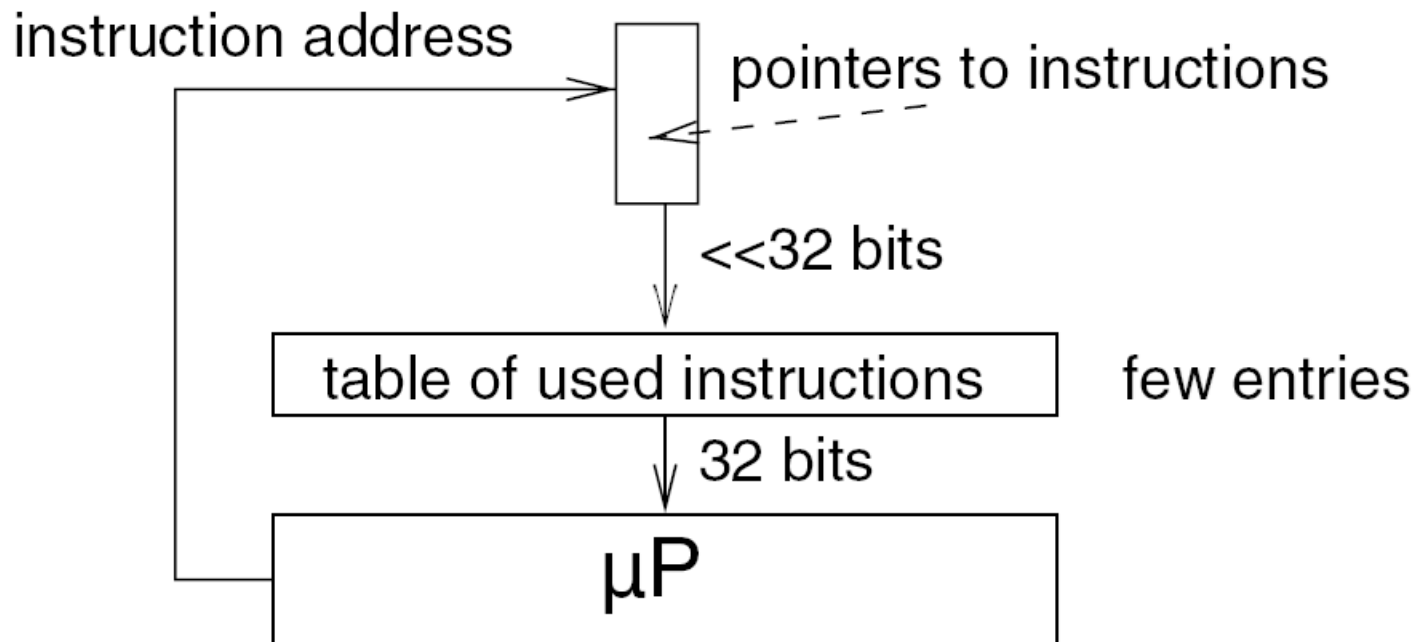
# Second Instruction Set

- **Example: ARM processor family**
  - ARM instruction set
    - 32-bit instructions
  - THUMB instructions
    - 16-bit instructions
- THUMB instructions are **dynamically converted** into ARM instructions.



# Dictionary

- Each instruction pattern is stored only once.
- A **look-up table** provides a **pointer** to the corresponding instruction in the instruction table (Dictionary).



# Example

```
1    xor ax,ax
2    add ax,2
3    mov cx,4
4  L1: add bx,ax
5    add ax,2
6    sub dx,ax
7    add ax,2
8    loop L1
9  L2: xor ax,ax
10   mov cx,4
11   add ax,2
12   loop L2
```

```
1    xor ax,ax
2    add ax,2
3    mov cx,4
4  L1: add bx,ax
5    2
6    sub dx,ax
7    2
8    loop L1
9  L2: 1
10   3
11   2
12   loop L2
```

```
1    xor ax,ax
2    add ax,2
3    mov cx,4
4  L1: 2 bx,ax
5    2 ax,2
6    sub dx,ax
7    2 ax,2
8    loop L1
9  L2: 1 ax,ax
10   3 cx,4
11   2 ax,2
12   8 L2
```