Computer Architecture

Example: Palindrome

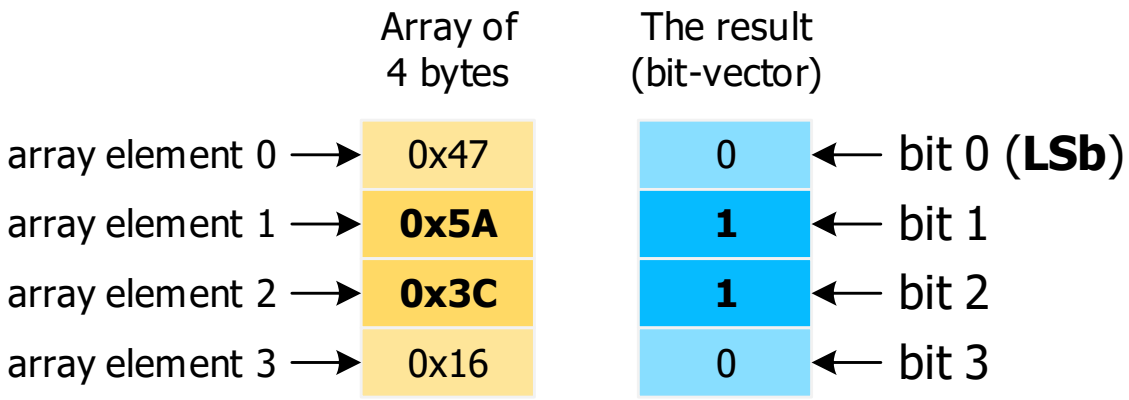
Fall 2017

Mirjana Stojilović

EPFL – I&C – PARSA

Problem Definition

- A **palindrome** binary number is a binary number that reads the same backwards as forwards. For example, the following two 8-bit numbers are palindromes: `01100110` and `10111101`.
- In this exercise, you are to write an assembly program that
  - reads an array of 8-bit numbers (array of **bytes**),
  - detects the bytes that are palindromes, and
  - creates a bit-vector in which a **bit** in the position **i** is **set only if** the **byte i** of the array is a palindrome. For an array of **N** bytes, the output bit-vector is **N-bit** long.



- The array starts at address **ARRAY**.
- The number of bytes in the array is always a multiple of 32 and it is stored in a 32-bit word located at address **ARRAY SIZE**.
- The resulting bit-vector should be stored in memory, starting from address **RESULT**.
- The values of these three symbolic constants are all word-aligned and smaller than **0xFFFF**.
- **Instructions**:
  1. Assume a **little endian** machine.
  2. To access the memory you are allowed to use only **load-word** and **store-word** instructions. These access 32-bit words.
  3. You may use either the MIPS or the Nios II assembly instruction set.
  4. Your code should conform to the assembly coding conventions.

a)

Consider the following array of 32 bytes (mem[ARRAY SIZE] = 0x20), with palindrome numbers appearing in bold:

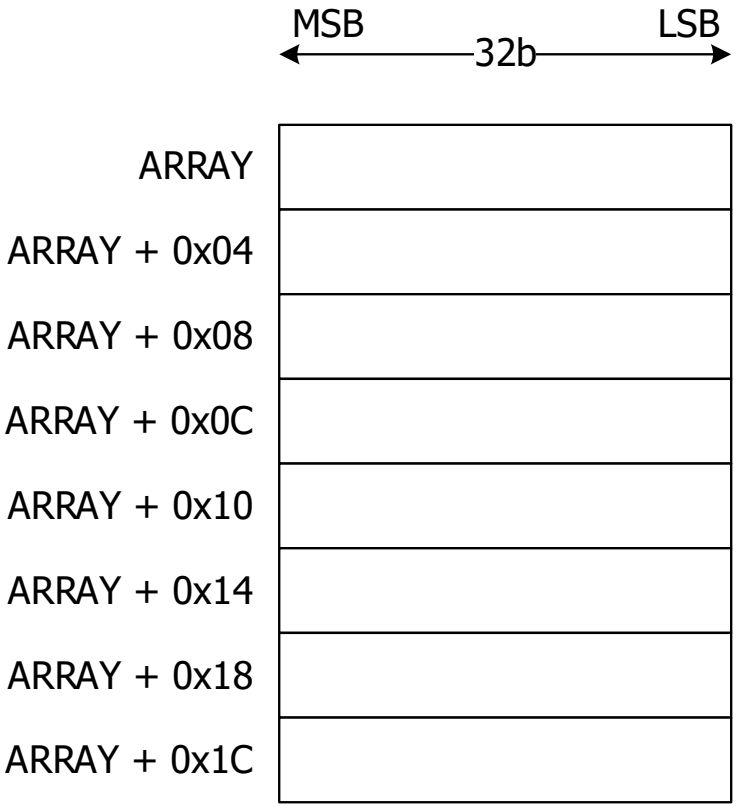
0x12, **0xFF, 0x3C**, 0x11, 0x54, **0x42**, 0xA0, 0xAA,

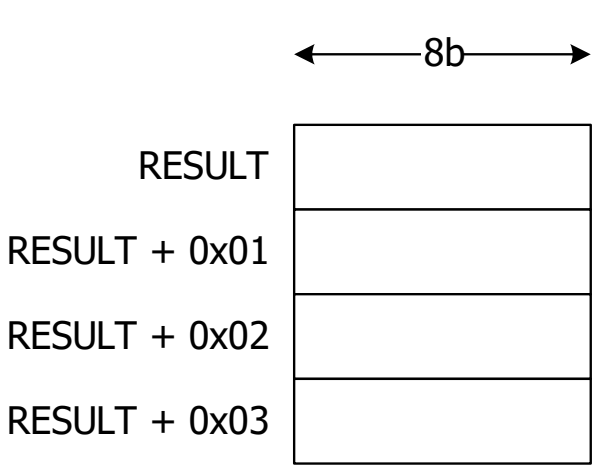
0x00, 0xDE, 0xAD, 0xBE, 0xEF, **0xA5**, **0x5A**, 0x13,

0x00, 0x00, ..., 0x00

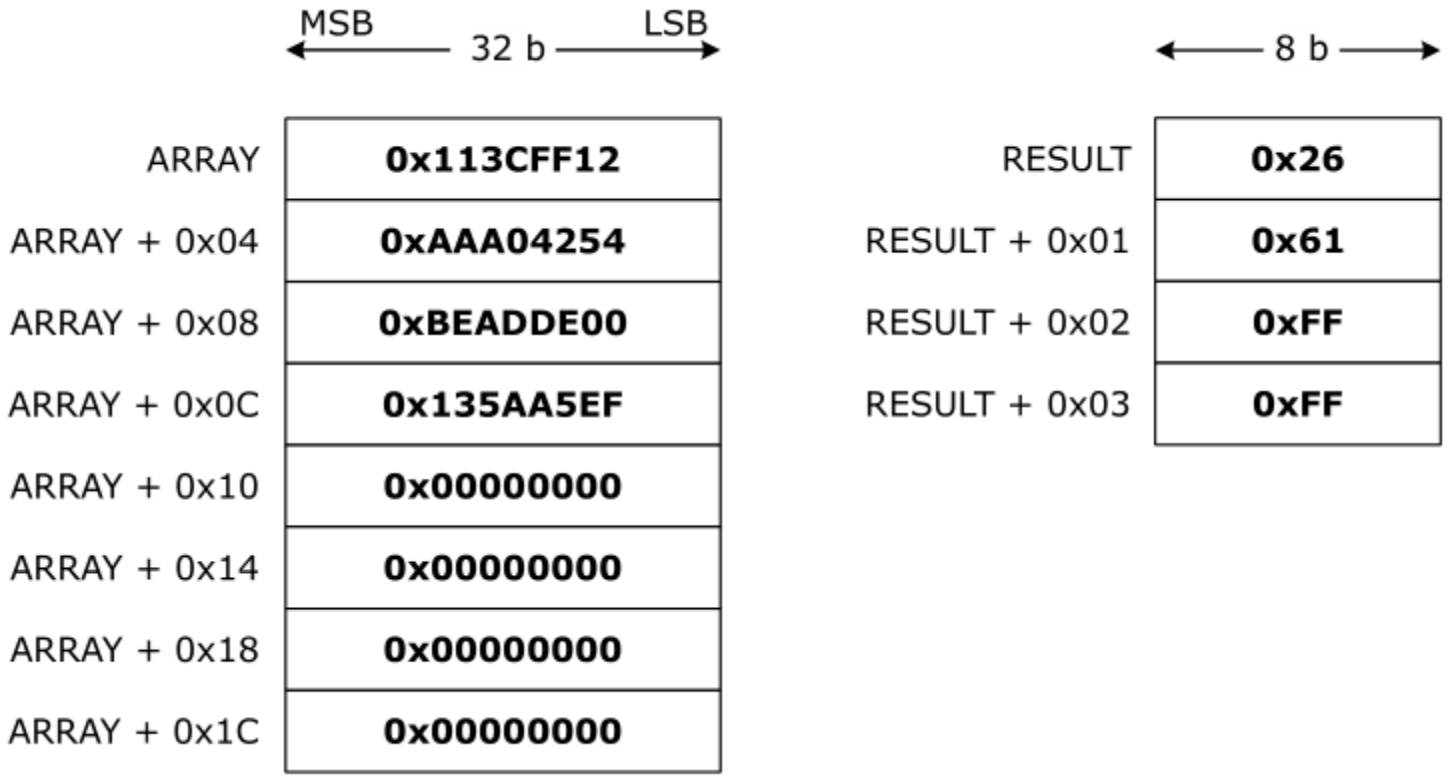
Show the content in the memory strictly respecting the following layout.

1. In the table on the left, show the content of memory from address **ARRAY** to **ARRAY + 0x1C**, assuming that the above sequence is stored in memory in the exact same order as above and starting from address ARRAY.
2. In the table on the right, show the content of memory from address **RESULT** to **RESULT + 0x03** after the execution of the program.





a) Solution

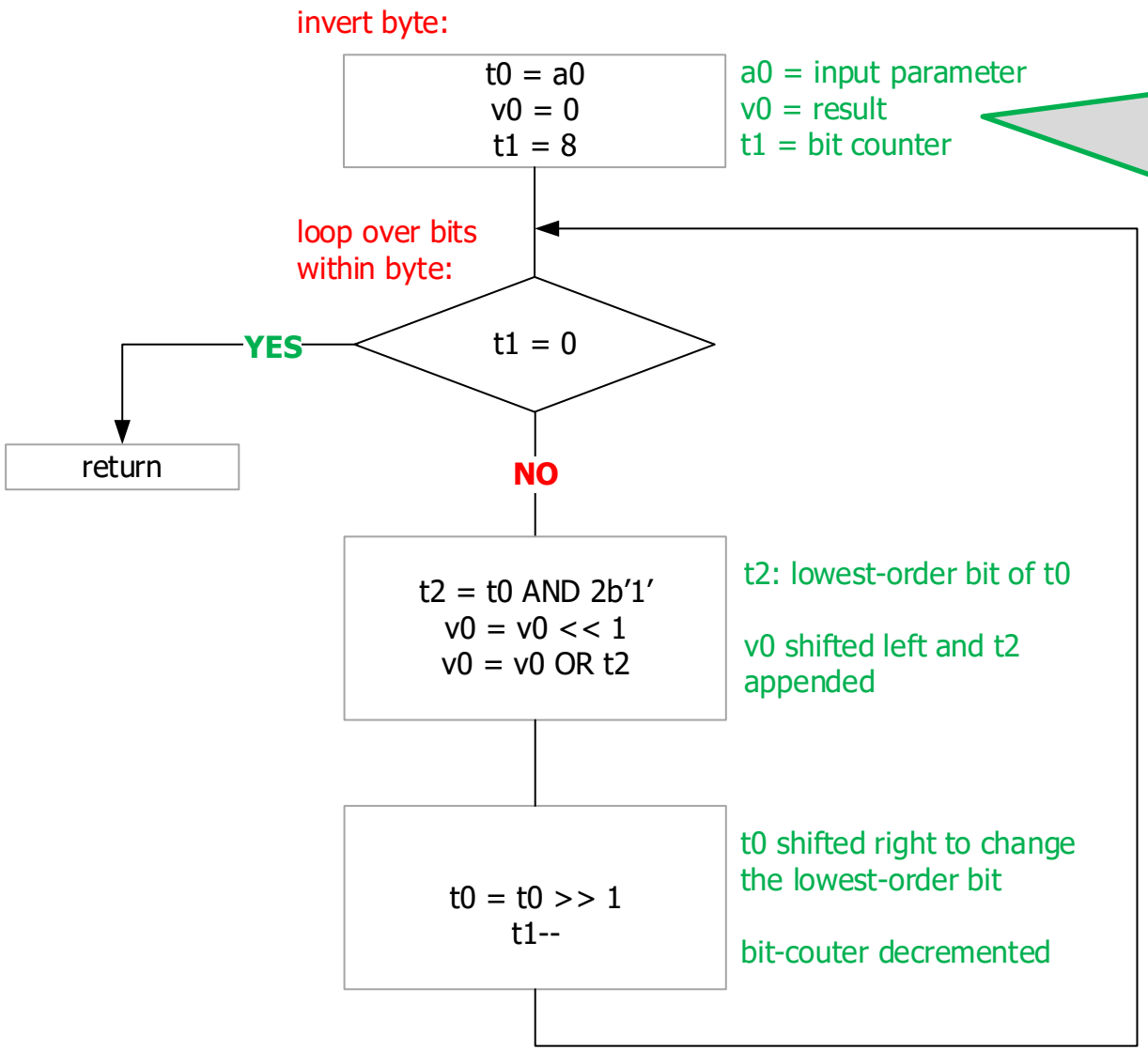

MSB
32 b
LSB
ARRAY
0x113CFF12
ARRAY + 0x04
0xAAA04254
ARRAY + 0x08
0xBEADDE00
ARRAY + 0x0C
0x135AA5EF
ARRAY + 0x10
0x00000000
ARRAY + 0x14
0x00000000
ARRAY + 0x18
0x00000000
ARRAY + 0x1C
0x00000000
8 b
RESULT
0x26
RESULT + 0x01
0x61
RESULT + 0x02
0xFF
RESULT + 0x03
0xFF

b)

An easy way to detect whether a binary number is a palindrome is to first reverse it and then compare it with the original binary number. If the original and the reversed binary numbers are the same, then the original binary number is a palindrome.

Write a function **`invert_byte`**, which takes as its argument an 8-bit binary number and returns this number **reversed**. For example, if the function argument is `00001011`, the function should return `11010000`.

Function invert_byte: Flow Diagram



According to the assembly coding convention, function **arguments** should be in registers **$a0–$a3** and function **return** values in registers **$v0–$v1**.

Additionally, the function arguments must be **preserved** on call; thus the use of temporary $t0.

c)

Write the program's main function, which traverses the array of bytes, detects if a byte is palindrome using the **`invert_byte`** function, and creates a corresponding bit-vector as described above.

Program: Flow Diagram