COMPUTER ARCHITECTURES (02LSEOV)

Prof. Montuschi

Problem solving session nº3
2017/2018

1. Write a program with an infinite loop. The register AX is initialized to zero before entering the loop, then it is incremented at every iteration of the loop.
   Suppose that the frequency of the clock signal received by the 8253 is 1 KHz. After one minute, the 8253 raises an interrupt. The 8086 immediately checks if the value stored in AX is a prime number. If so, the variable **isPrime** is set to 1, otherwise it is set to 0.

2. Configure Intel 8255 in mode 0 for group A, with port A in *output*. Realize a program that every 10 seconds writes a new value of the Fibonacci sequence on port A:
   0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233
   No more values are written after 233.
   Suppose that the clock frequency of the 8253 is 100 KHz.

3. Configure Intel 8255 in mode 0 for groups A and B, with ports A and B in *input*.
   Every 35 seconds, a new value is read from port A and stored in **rowArray**, which is an array of bytes with **N** elements. Similarly, every 27 seconds, a new value is read from port B and stored in **columnArray**, which is an array of bytes with **M** elements. **N** and **M** are constant defined by the programmer. Suppose that the clock frequency of the 8253 is 1 KHz.
   Write the interrupts service routines for acquiring data from the 8255 and filling the arrays. When the first array becomes full (depending on the values of **N** and **M**), new values read on the corresponding port of the 8255 are discarded while the other array is still being filled. When both arrays become full, the 8086 computes the product of the two arrays as soon as it receives the first interrupt request from any counter of the 8253. The product is stored in **matrix**, which is a matrix of words with **N** * **M** elements.
   The product between a column array and a row array is:

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} (y_1 \quad y_2 \quad \cdots \quad y_n) = \begin{pmatrix} x_1y_1 & x_1y_2 & \cdots & x_1y_n \\ x_2y_1 & x_2y_2 & \cdots & x_2y_n \\ \vdots & \vdots & \ddots & \vdots \\ x_my_1 & x_my_2 & \cdots & x_my_n \end{pmatrix}$$

4. Configure Intel 8255 in mode 1 for group A and mode 0 for group B. Port A is in *input* and port B is in *output*.
   A sequence of byte is received from port A and stored in the array **inputSequence**. Every byte corresponds to an unsigned integer. The sequence has 20 characters at most and it ends when values ‘0’ is received.
   When the full sequence is received, the program writes on port B, every 5 s, the values of an output sequence. The output sequence is obtained by filtering **inputSequence** as follows:
   a. every value of **inputSequence** lower than the half of the maximum value is removed from the output sequence
   b. consecutively repeated values in **inputSequence** are written only once.

   Example:
   Input sequence 1 27 27 3 45 45 45 27 0
   Output sequence 27 45 27 0
   Values 1 and 3 are removed because they are lower than maximum/2 (which is 45/2).