Chapter 2

Introduction to myCPU

The myCPU is a fully modular **8-bit TTL/CMOS CPU** designed using only discrete logic components and implemented with the most common integrated circuits from the 74xx family over CMOS or TTL technologies. The myCPU project uses the **Microprograming Technique** to process instructions, relying on **Microinstructions** the instruction execution flow. The myCPU supports debugging at microinstruction level. It was designed on printed circuit boards (PCBs) and it can built using TTL or CMOS technologies.

The feature of debugging at microinstruction level allows to observe the execution of individual microinstructions, stepping through each one and view the state of all logic components statically. This provides a detailed view of the instruction execution flow in real time.

2.1 A development and learning Platform

The myCPU was designed as a learning platform to provide an educational experience in the workings of a CPU, including the logic elements involved in its basic architecture and how several digital logic elements interact in a synchronized way.

With a modular design and hardware-level accessibility, myCPU serves as development platform. The myCPU platform enables the direct access to the relationship between hardware and software and give builders the opportunity to design their own modules. Thanks to the Instruction Decoder module, a programmable EEProm-based microinstruction decoder, builders can create their own instruction set for program writing.

The myCPU architecture was designed with the purpose to provide a clear understanding of a CPU system and its workings to people of all levels, including beginners.

Table 2.16: myCPU Planned modules

Description
BUS module with support to hybrid microprogramming
Variable speed Clock module from 1 to 2Mhz
Improved ALU based on 74LS181
ALU Shifter, Rotator and Bit manipulation module
Registers 16 bits with LOW/HIGH load
Registers 8 bits with LOW/HIGH out
MAR 16 bits, capable to addressing up to 64K bytes of memory
SRAM module up to 2K based on 6116 family
SRAM module up to 64K based on 62256 family
SRAM Core Memory 64 x 1 bits
Program Counter 16 bits
Sequencer up to 16 steps
Flags Register up to 8 flags
Input register 8 bits
Output Register up to 16 bits
Stack Pointer 8 bits
EEProm Boot module to load program into SRAM at start up to 64K
Instruction Register 8 bits with support up to an 256 Opcodes
Instruction Decoder with decoding support up to 14 bits based on WD27010
Instruction Decoder with decoding support up to 17 bits based on WD27020