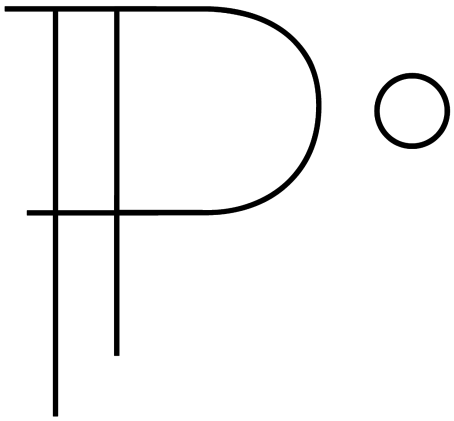


PHANAND ONE
Datasheet

Introduction

The PHANAND ONE is the first computer and 8-bit CPU made by Enrique Phan. It is homemade and made only using discrete logic, specifically the 74XX series chips.

It features a specific architecture, in the middle between a Von Neumann and a modified Harvard architecture, where two different memory devices can hold both instructions and program data.

Apart from the CPU itself, it has built-in support for SPI communication and a PS/2 keyboard interface through an “input register”, which supports interrupts. To provide an output, it also features a VGA controller with two different VRAMs to allow different program flows.

The CPU was designed to be flexible but simple at the same time. Therefore, it has four general purpose registers(GPRs) to make the necessary operations with minimum memory access and four address registers(ARs) connected to both the RAM and EEPROM devices. All four GPRs are connected to the ALU, meaning it can take as first or second operand any of of them.

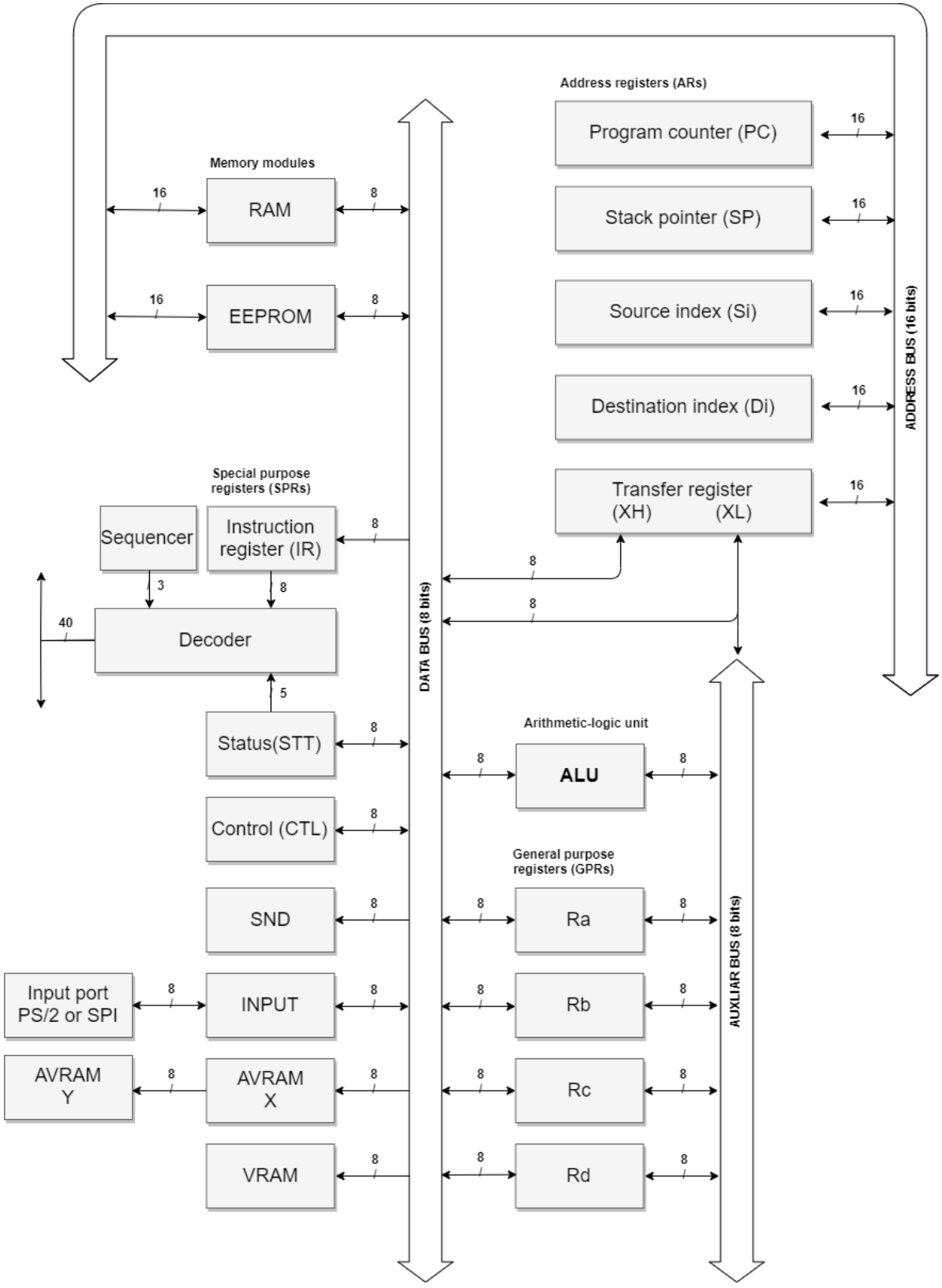
Apart from that, special purpose registers(SPRs) exist to control the CPU or to interface with the I/O modules, as mentioned earlier.

The architecture and each register will be carefully described in the next pages.

Features

- Up to 2 MHz stable clock rate
- 64kB RAM + 64kB ROM
- Four general purpose registers
- Four address registers
- Flexible and complete instruction set, allowing different addressing modes and permutations.
- Interrupts: Keyboard and boot(reset)
- Complete ALU, supporting different arithmetic and logic operations, including logical shifts
- Built-in VGA controller, featuring 2x64kB VRAMs and hardware support for shared access between the CPU and the controller.
- SPI interface, specifically designed to work with PS2 controllers, although it can work flawlessly as a standard SPI bus.
- Interchangeable interrupt vectors (hardware)
- Easy to reprogram microcode, allowing the user to make new instructions or modify already existing ones.
- Discrete logic design (74XX series chips)
- Special architecture, where both memory devices can hold instruction and program data
- Output register to control a user desired device, like a sound module.
- More than 200 LEDs to see the program flow when running on stepped mode or low clock rates.
- Stepped CPU with variable machine cycles.
- Status and control registers which allow the user to set up the CPU, or the CPU to inform the user about the machine state.

Architecture overview



Special purpose registers

Status register (STT)
R/W

D7	D6	D5	D4	D3	D2	D1	D0
IMG	#BF	BSY	LF	VF	NF	ZF	CF

On the one hand, the status register holds the ALU flags output from the last operation. On the other it controls crucial parts of the CPU. It is important that all bits are read/write

-**IMG:** Allows the user to check if the last VRAM access was successfully done when sharing access, meaning the CPU is writing to the same VRAM which is currently being read from to output the image. If ‘1’, last VRAM access by the CPU was not successful and should attempt the write again. If ‘0’, last VRAM access was probably a successful write.

It should be manually set to LOW after every read, since the video controller can only set the bit high.

-**#BF:** Boot flag(active low) selects the instruction fetch memory device. It is reset on power up. If ‘0’, the instructions are fetched from ROM. When set to ‘1’, the RAM module becomes the main memory device, leaving the ROM accessible through special instructions.

-**BSY:** Enables or disables interrupts. When ‘1’ the system does not allow for new interrupts. When ‘0’, interrupts are allowed. Disabled on power up(LOW). After every interrupt, it is set to ‘1’ to prevent the machine from being constantly interrupted before finishing the interrupt service routine(ISR)

-**Flags:** D4 down to D0 hold the CPU flags from the last ALU operation, although the user can write to them, this is useful for pushing the flags on to the stack.

- -LF: Logical carry flag
- -VF: Overflow flag
- -NF: Negative flag
- -ZF: Zero flag
- -CF: Carry flag

Control register (CTL)
R/W

D7	D6	D5	D4	D3	D2	D1	D0
ACK	SCK1	SCK0	SS1	SS0	IS	VR1R	#VR1W

Defines different CPU and computer behavior

-**ACK:** Similar to the IMG bit from the status register, it is set to ‘1’ when the PS2 controller generates the ACK signal. It should be read to ensure reliable and fast communication and reset every read. It can be ignored when using the standard SPÎ protocol.

-**SCK:** D6 and D5 select the serial clock for the SPI interface by dividing the main clock.

D6D5 = 0b11	D6D5 = 0b10	D6D5 = 0b01	D6D5 = 0b00
CLK/4	CLK/8	CLK/16	CLK/32

-**SS:** D4 and D3 select the slave in the SPI bus

D4D3 = 0b11	D4D3 = 0b10	D4D3 = 0b01	D4D3 = 0b00
Standard	Controller 2	Controller 1	Idle(no comm)

Standard means it is using the specific external SPI port, where the user may connect a device to communicate with

-**IS:** Input select decides which data is read from the input register. If ‘0’, keyboard data is read; if ‘1’, controller data is read

-**VR1R:** Selects VRAM to be read from by the VGA controller. When ‘1’, VRAM1 is read, when ‘0’, VRAM2 is read.

-**#VR1W:** Selects VRAM to be written to by the CPU, When ‘1’, VRAM2 is written, when ‘0’, VRAM1 is written