**TICK – An 8-bit Experimental Bit Serial Computer**  
  
Tick is an experimental, 8-bit, bit serial computer designed to interface to SPI peripherals and memory.

It has a 16-bit Program Counter, a 16-bit B-Register, an 8-bit Accumulator and an 8-bit Input Register.

The circuit may be subdivided into several sub-sections for easier analysis.

1. Clock Sequencer and Timing Pulse Generator (TPG)

2. Instruction Decoder

3. Program Counter (PC)

4. Arithmetic and Logic Unit (ALU)

5. Conditional Branch Logic

6. Memory Access

7. Input/Output (I/O)

8. ROM Emulation

**1. Clock Sequencer and Timing Pulse Generator (TPG)**

This is at the heart of the design and co-ordinates all of the control signals.

A 14-bit binary counter U6 (74HC4060) provides a crystal oscillator with a 16MHz crystal Q1. This is divided down to provide 10, link selectable clock outputs from 1Khz (approx) to 1MHz. The selected clock feeds a 4-bit counter, U8 (74HC161) and this is used to generate a burst of 16 clock pulses for the Program Counter (PC-CLK) and 8 gated clock pulses (GCLK) for the 8-bit shift registers. A SR latch formed around NAND U7 (74HC00) with triple input AND U11 (74HC11) provides a means of gating the clock to generate PC-CLK and GCLK.

The terminal count of the 74HC161 (RCO) is used to feed a chain of 4, D-type flipflops U9 (74HC175), and these "top and tail" pulses T0, T1, T17 T18 and their complements are used to synchronise the loading of shift registers andread/write access to the SRAM.

A power-on reset circuit is formed around U19A and U19B. This resets all shift registers and counters when power is applied to the circuit. There is also a RESET switch S13, which allows a manual reset to be performed.

**2. Instruction Decoder**

The instruction from the ROM is 16-bits wide and consists od an instruction field IR7:0 and a data field P7:P0.

The instruction field is further divided into a 4-bit operand IR7:IR4, and a condition field IR3:IR0.

If IR7=0, the remainder of the 16-bit word is treated as a literal and is loaded into the B-Register.

If IR7=1, this indicates a RAM memory reference instruction with IR6:4 defining the operation, and the remainder of the instruction word IR3:0 and P7:0 treated as a 12-bit address which is output onto the RAM address bus MA11:MA0.

Instruction decoding is done with 4 or 5 "basic gate" ICs. As entirely combinational logic, it could also be implemented in a small ROM, or even with a diode matrix.

Instruction decoding begins with U10 (74HC138) a 3 to 8 line decoder, which performs the primary stage of decoding.

Here the instruction field IR6:4 is decoded into the 8 instruction groups, with one of its "Y" outputs being set low for each individual instruction group - as follows.

/Y0 LOAD

/Y1 AND

/Y2 XOR

/Y3 OR

/Y4 ADD

/Y5 SUB

/Y6 STORE

/Y7 JUMP

These low going outputs are further combined in logic U11, (74HC11), U12 (74HC86) and U13 (74HC00) to provide a series of control signals, time selected by the timing pulses T0, T1, T17 and T18.

Quad 2-input NAND gate U13 does the bulk of the secondary decoding, where instruction groups /Y0 to /Y7 are gated with timeslot pulses.

The two XOR gates U12B and U12C have one input tied high and thus act as simple inverters. U12A is also an inverter and is connectedas an 8-input wired OR gate with 8 input diodes and a pull-down resistor, used to detect when the output of the Accumulator or memory on the data bus is equal to zero.

U11 is a triple 3-input AND. U11A detects if any of the Boolean instructions have been selected (/Y1, /Y2, /Y3) and is used to suppress the carry.

**3. Program Counter.**

This is a 16-bit counter and consists of two 74HC595 serial in parallel out (SIPO) shift registers U2 and U3 connected in series. As a bit serial computer, it is reasonable to use shift registers and a half adder to create a Program Counter, especially as it can be loaded from a serial data stream.

The registers convert a serial bitstream into a parallel address, which is used to address the 64K x 16-bit ROM (U1 AT27C1024). For development purposes, the ROM is replaced with a Pi Pico clone, for a more efficient development cycle.

The 74HC595 is a SIPO shift register, with all outputs connected to an octal tristate latch. Once the shift register contains the required address, this can then be latched onto the output pins, where it will remain for the whole duration of the machine cycle. In this respect, the PC "freezes" the address and as such acts as the Instruction Register, holding the Instruction for the complete Fetch, Execute, Writeback cycle. As the lower 12-bits of the ROM contain the effective memory address (EA), the address of the operand read, will be retained for the writeback operation.

A Program Counter has two modes of operation, either to increment the ROM address by 1 at the end of each machine cycle, or to load the shift registers from an external serial source to allow a jump to be executed.

The PC increment hardware is based around a half adder, followed by a 2-input multiplexer. The mux chooses between the increment mode and the load (jump) mode.

The PC half adder is provided by one half of a 4-bit full adder (74HC283 U14). The multiplexer is one half of a 74HC126, quad tristate buffer (U15). The other halves of U14 and U15 are used in the ALU, so not wasted. Instruction group 7 "Jump" is used to swich the PC multiplexer between its +1 increment mode and the jump mode, where it is loaded with data from the B-Register, which has been loaded from the ROM or from and external source.

**4. Arithmetic and Logic Unit (ALU)**

The ALU is based around a full adder, in this case provided by U14 (74HC283) four bit adder that has been partitioned to provide a half adder for the Program Counter and a full adder for the ALU. The full adder can perform subtraction too, by inverting the input from the B-register and adding 1, by way of the Carry input.

The full adder provides sum (XOR) and carry (AND) outputs, which are selected using a 2-input multiplexer, provided by U15 (74HC126). This allows the ZERO, AND, XOR and OR bitwise Boolean logic functions to be selected by way of two instruction control lines I0 and I1. If a Boolean function is selected, the carry propagation from bit to bit is suppressed.

**5. Conditional Branch Logic**

During a conditional branch, the 4-bit instruction field IR3:0 defines the condition as follows by testing the ZERO flag and the MSB of the Accumulator:

0000 Long Jump

0001 AC > 0 BGT

0010 AC < 0 BLT

0011 AC != 0 BNE

0100 AC = 0 BEQ

0101 AC >= 0 BGE

0110 AC <= 0 BLE

0111 Branch Always

1xxx

A 4 -input multiplexer, U23 (74HC153) receives the condition field IR2:0 on inputs C2:C0. The JUMP signal is used to enable the multiplexer via pin 1.

The ALU status bits ZERO and SIGN (BUS7), are applied to select inputs A and B. If a condition is met the BCC output on pin 7 is raised.

The other half of the multiplexer is not used.

The BCC output is combined in NAND U28 (74HC00) to provide a LONGJUMP control signal if IR3 = 0. If IR3 = 0 the jump address is taken from the 11 lower bits of the instruction word, limiting it to 0 to 2047 in RAM.

**6. Memory Access.**

The 32K byte RAM U18 (62256) is read onto the data bus during timeslot T0, and written back to during timeslot T18.

The address for the RAM comes from the full 16-bit output of the Instruction ROM but with the exception that IR7=1, so a range of 32768 addresses may be reached.

There are 2 distinct forms of Load instruction:

IR7=1 Memory Reference Instruction

IR3:IR2 and P7:0 forms a 12-bit address for the A operand to place in the Accumulator. This applies to LOAD, AND, XOR, AND, ADD, SUB and STORE instructions.

IR7=0.

Load Accumulator with data from address IR7:0 and P7:0 thus covering addresses 0 to 32767, which is the full extent of the fitted RAM.

Load the B-Register with the 15 bit constant, IR6:0 and P7:0

**7. Input/Output (I/O)**

TICK is intended to work with SPI peripherals and memory. As such it implements a basic SPI bus with MOSI, MISO, SCLK and /SS signals.

The serial output of the Accumulator U17 (74HC299) is fed through a tristate buffer U24A and sends this MOSI signal to connector X1 pin 2. The MISO signal is received by U22 (74HC595) 8-bit tristate, latchable shift register on X1 pin 3 and can be gated onto the 8-bit data bus. The serial clock X1 pin 4 is just the 8 cycle gated clock burst, GCLK. The slave select /SS or /CE (chip enable) is the inverted output Q4 of the gating counter U8 (74HC161). A maximum SCLK of 1MHz can be achieved from the current 16MHz clock crystal oscillator.

**8. ROM Emulation and Peripheral Functions**

Whilst TICK was designed to have firmware coded into a 64K x 16-bit ROM, U1 (AT27C1024) it makes sense to emulate the ROM with a small microcontroller - in this case a low cost clone of the popular Raspberry Pi Pico module.

The Pico plugs into 2 x 20 way female pcb socket J3 on the left hand edge of the pcb. It's GPIO lines 0 to 15, "sniff" the ROM address generated by the Program Counter, and then look up the associated instruction word from a table of ROM contents held in the flash memory of the Pico. This 16-bit word is then output to a pair of 74HC595 shift registers, U26 and U27 located below the Pico module, which present the "ROM" data to the relevant ROM data lines IR7:0 and P7:0.

The Pi Pico provides regulated 3V3 supply for the TICK board and also a USB serial port for debugging. Unused GPIO, ADC chanels and other resources on the Pico can be used to support the TICK board. The Pico can act as a smart serial peripheral and connect to the TICK via the SPI port connector X1.