

Jon Stocker/Ian Shaw

## How to Use the TRAC Support Chip

### Programming TRAC

#### **Structure of TRAC**

Each TRAC cell can be configured to one of eight functions. Inside TRAC, digital registers are used to set each cell to the appropriate function. Three bits are needed to represent the eight options, and so each cell has a 3-bit register. These registers are cascaded together to allow serial data to 'ripple' through the TRAC chip.

The serial programming code, (a stream of logic '0's and '1's) enters the device at the 'Data' Pin and travels through each register controlled by the clock signal. A transition from logic 0 to logic 1 on the 'Clock' Pin causes each data bit to be shifted from one register bit to the next.

Essentially the clock signal is a square wave (of arbitrary duty cycle, within limits) with a number of positive edges matching the total number of register bits. That is, the number of TRAC cells multiplied by three.

The time taken to program a device is the total number of register bits multiplied by the clock period;  $1/\text{clock frequency}$ .

#### **Synchronisation**

Serial data can be accepted at any speed up to the maximum clocking frequency but it has to be 'in step' with the Clock signal. In practise this means the bit of data to be received by TRAC must be there before the Clock goes high *and* must have been replaced by the next data bit before the clock goes high again.

If this is not the case, one bit may be read twice, or if data bits are replaced before being read, they will be lost. Obviously this will result in a corrupted design.

### **Multiple TRAC devices**

If the *same* design is needed in more than one device, they can be programmed in parallel. In this case all DATA pins are connected together and all CLOCK pins are connected together. Any number will take the same time to program as one individual device.

The number of devices will be limited in practice by the electrical drive capability of the clock and data sources, this should be a few tens of devices at least with the devices recommend in this text. As the registers are CMOS, the load presented to the signal sources will be a function of frequency.

If two or more devices are required to have different designs, which is more often the case, the devices are serially programmed. The CLOCK pins are connected together as above, but now the DOUT pin of the first device is connected to the DATA pin of the next device and so on. The DOUT pin of the last device is not needed.

It is important to remember that as the code for the last TRAC device has to flow through all the other devices first, it must be sent *first*. In the same way, for each TRAC design, the code of the last cell (bottom right of design in TRAC software) must be sent first.

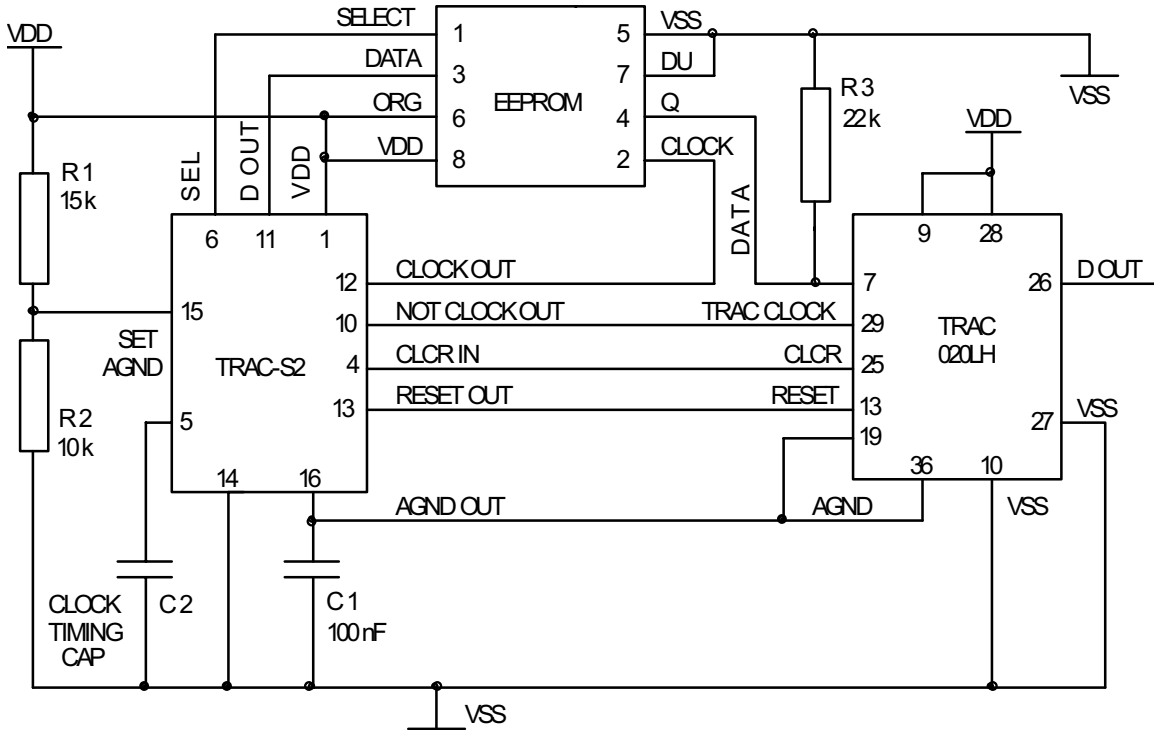
The time taken to program serially will be multiples of the individual time. This time the practical limit on device numbers is dependent on the drive capability of the clock signal only, as each device's DATA pin is driven by the last's DOUT pin. There is no summation of propagation delays because the clock synchronises data flow.

## **The TRAC Support Chip**

### **Introduction**

For applications that do not use a microcontroller, the required data can be stored on a serial EEPROM, such as the ST9346CB1. The function of the TRAC support chip is to get the data from the EEPROM into the TRAC device's shift registers on power up. In addition the TRAC Support chip's adjustable 'rail splitter' provides a low impedance signal ground rail. The integrated supply monitor initiates the program load on power up and after power failure or serious fluctuation.

## Connection Diagram and Pin Description



| Components                  | Description / Notes               |
|-----------------------------|-----------------------------------|
| TRAC-S2Q16                  | TRAC Support Chip                 |
| ST9346CB1                   | EEPROM                            |
| TRAC020LHQ36                | Pin out for 36 Pin QSOP TRAC020LH |
| R1                          | 15 k $\Omega$                     |
| R2                          | 10 k $\Omega$                     |
| R3                          | 22 k $\Omega$                     |
| C1                          | 100 nF                            |
| C2 (Clock Timing Capacitor) | Between limits 47 pF & 100 nF     |

### TRAC Support Chip - TRAC-S2Q16

| Pin Number | Name            | Description  |
|------------|-----------------|--|
| 1          | V <sub>DD</sub> | 5.0 V Power supply   |
| 4          | CLCR_IN         | <b>Input</b> - Active high (5.0V) enables the CLOCK when RESET_OUT is logic high (5.0V)  |
| 5          | CAP             | <b>Input</b> - Capacitor to V <sub>SS</sub> sets the clock frequency   |
| 6          | SELECT_OUT      | <b>Output</b> - Switches active high (5.0V) on DATA falling edge after RESET is switched active high (5.0V). This enables the EEPROM |
| 10         | NOT CLK_OUT     | <b>Output</b> - Inverted CLK_OUT used to clock the TRAC devices  |
| 11         | DATA_OUT        | <b>Output</b> - CLK_OUT divided by four, provides the EEPROM instructions to read its data from location 25                          |
| 12         | CLK_OUT         | <b>Output</b> - used to clock the EEPROM   |
| 13         | RESET_OUT       | <b>Output</b> - Resets the TRAC data shift registers to logic low (0.0V)   |
| 14         | V <sub>SS</sub> | 0.0V Power supply  |
| 15         | SETV            | <b>Input</b> - Defines the AGND_OUT voltage. Set by potential divider  |
| 16         | AGND_OUT        | <b>Output</b> - Analogue Ground for TRAC. This is able to sink/source current  |

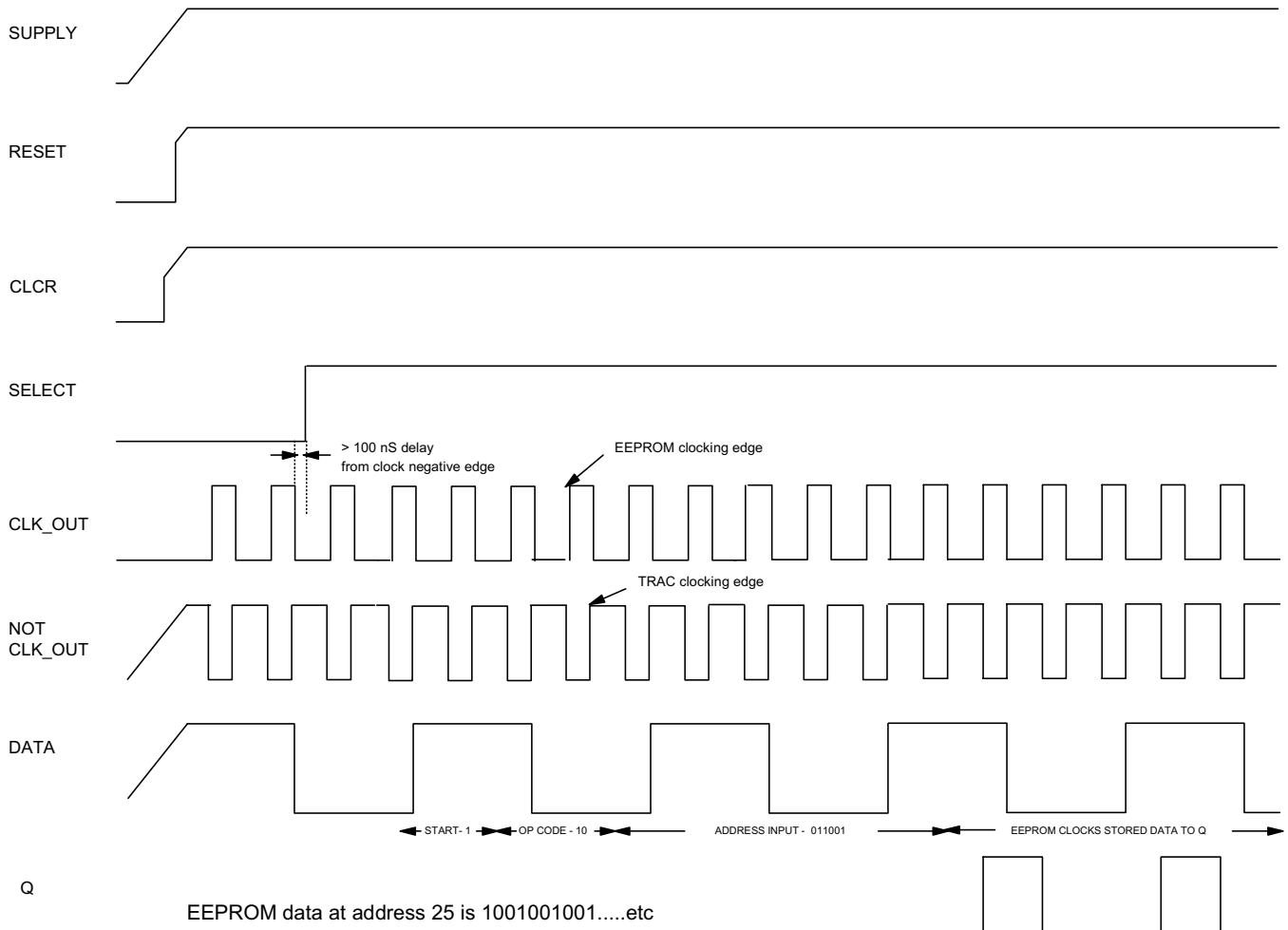
### EEPROM - ST9346CB1

| Pin Number | Name   | Description  |
|------------|--------|--|
| 1          | SELECT | <b>Input</b> - chip select, the EEPROM will only output data when this pin is set to logic high (5.0V)               |
| 2          | CLOCK  | <b>Input</b> - Clock input from the TRAC support chip  |
| 3          | DATA   | <b>Input</b> - Data input, this is the instruction set for the EEPROM to read it's stored                            |
| 4          | Q      | <b>Output</b> - The output for the stored data   |
| 5          | VSS    | 0.0V Power supply  |
| 6          | ORG    | <b>Input</b> - Organisation select, when connected to V <sub>DD</sub> the EEPROM is configured as 64 words x 16 bits |
| 7          | DU     | Don't Use - Recommend connect to V <sub>SS</sub> for lowest standby current  |
| 8          | VDD    | 5.0 V Power supply   |

### TRAC Device - TRAC020LHQ36

| Pin Number | Name  | Description   |
|------------|-------|---|
| 7          | DATA  | <b>Input</b> - Serial data input  |
| 9/28       | VDD   | 5.0 V Power supply  |
| 10/27      | VSS   | 0.0V Power supply   |
| 13         | RESET | <b>Input</b> - Set to logic low (0.0V) resets all the data shift registers to logic low (0.0V). The cells are then programmed to OFF.   |
| 19/36      | AGND  | Analogue ground for the TRAC device   |
| 25         | CLCR  | <b>Output</b> - Used as a control pin, this is an additional shift register connected to the last programming shift register and has a Q bar output (i.e. it resets to logic high (5.0V)) |
| 26         | DOUT  | <b>Output</b> - Serial data output from the last programming shift register. this allows the TRAC devices be programmed together  |
| 27         | CLOCK | <b>Input</b> - Used to clock the serial data to program the TRAC  |

## Timing Diagram



## Circuit Function

When the circuit is switched on, the  $V_{DD}$  supply voltage will ramp up to its nominal value of 5.0 volts. While the supply is ramping up, RESET\_OUT on the support chip is held logic low (0.0V). While the TRAC RESET is held logic low all the TRAC program shift registers will reset to logic low (all cells in OFF state). As its output is from Q bar the CLCR on the TRAC device is reset to logic high (5.0V). With CLCR\_IN logic high, the clock on the TRAC support chip would normally be enabled, but because RESET\_OUT is logic low during the supply ramp up, the clock is disabled. It is only when the supply reaches the threshold voltage of around 4.5 volts that RESET\_OUT switches to logic high. This then enables the clock on the support chip. RESET\_OUT will stay logic high so long as the supply voltage stays above the threshold voltage.

Once the clock is enabled CLK\_OUT will start switching from its start up level of logic low, while NOT CLK\_OUT does the inverse. On the second negative edge of CLK\_OUT, DATA\_OUT switches to logic low. This negative edge switches the latch on SELECT\_OUT, which then goes logic high. To satisfy the EEPROM requirements SELECT\_OUT goes logic high >100 nS after CLK\_OUT switched to logic low

With SELECT logic high the EEPROM is enabled, and DATA will now be clocked into the device on the positive edge of CLOCK. As DATA to the EEPROM from the support chip is CLK\_OUT/4. This gives DATA as 001100110011001100 etc. Ignoring the first two '0s' the EEPROM decodes the first '1' as the 'start', followed by an 'Op code' of '10' which sets the EEPROM into 'read mode'. The next six bits, which will be 011001, denote the read out starting address for the EEPROM. (memory location 25). The TRAC programming data must begin at this location or some higher location. The EEPROM after it has received the starting address ignores DATA, which will continue until the TRAC devices are programmed. The EEPROM then starts to read out the TRAC program data from the defined location. The TRAC software loads data to the EEPROM at location 25 by default. During read out this address is automatically incremented.

The TRAC program data is clocked into the TRAC device on the positive edge of NOT CLK\_OUT. Using this as the clock eliminates problems associated with logic race conditions.

The TRAC program data should be preceded by a '1'. This is used as a flag to stop data being clocked through the TRAC program shift registers when it has reached its correct location and this occurs when the '1' is clocked as a '0' at CLCR. When this is fed back to CLCR\_IN of the support device it disables the clock, leaving the TRAC chip with the required program.

If, for example, the TRAC chip was to be programmed with all NIP functions (for which the code is '100'), then the code read out from the EEPROM would be the initial '1' plus multiples of 001. (Note LSB first). The code would therefore be 1 001 001 001 001....001. (Spaces have been included for clarity and do not exist in the real code). The 001 is repeated twenty times for one TRAC chip resulting in 61 bits.