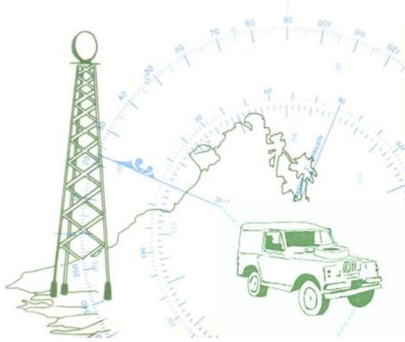
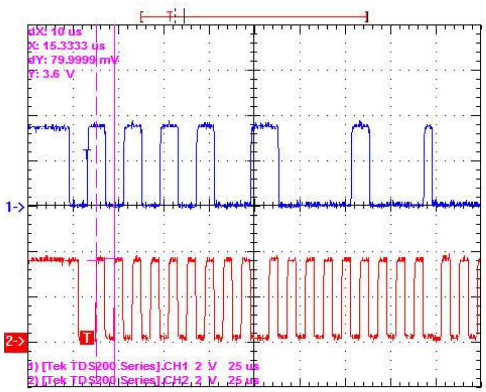
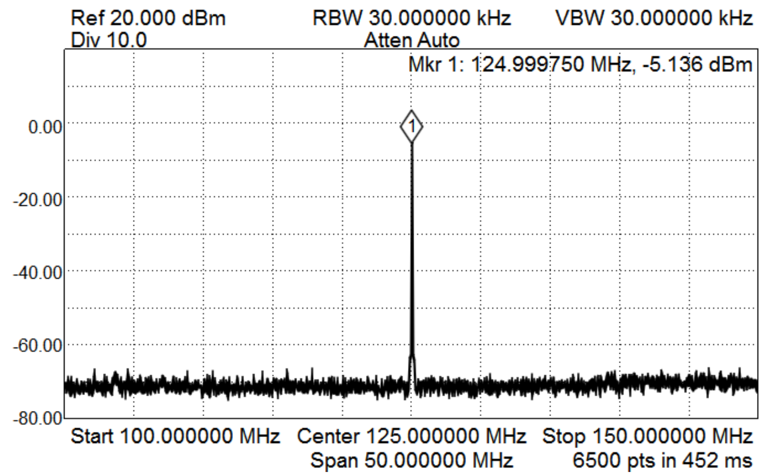


# Programming the Silicon Labs Si514 100KHz - 170MHz I2C XO with Arduino & Raspberry Pi




**Jeremy Clark VE3PKC**



## Copyright Information

ISBN 9780988049055



Clark Telecommunications
Jeremy Clark
500 Duplex Suite 506
Toronto M4R-1V6, Ontario, Canada
416-488-5382
jclark@clarktelecommunications.com
www.clarktelecommunications.com

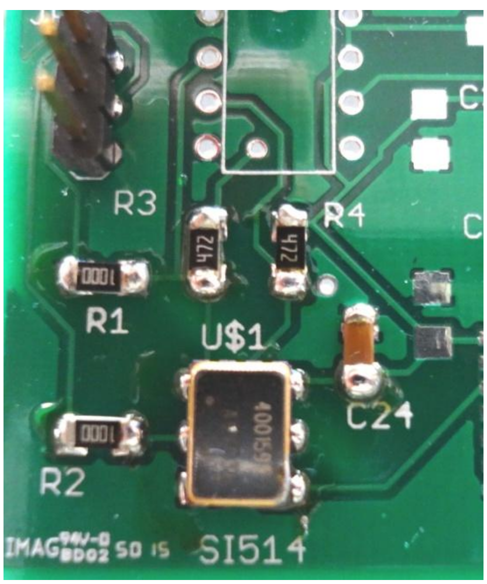
# Table of Contents

# 1 - Introduction

## 1.1 - Si514 I2C Programmable Oscillator

The Si514 (Ref.1) is a compact programmable oscillator with accuracy ideal for an amateur radio VFO or for data clocking applications. It comes in several voltage and speed options and 2 package sizes. Figure 1.1 shows the larger package size 5x7mm mounted on a PCB with 2 series protection resistors R1 & R2 on SDA/SCL and 2 pull up resistors R3 & R4, plus a bypass capacitor C24 on Vcc. The Si514 is programmed using the I2C bus. This publication is based on the 514CBB000112AAG unit which is CMOS, +/-50ppm, 100KHz - 170MHz, 10MHz startup frequency, $55 I2C address, 5x7mm package. The part number convention is shown in Figure 8 page 28 of the data sheet.

| **Si514 Applications** | Frequency Resolution = 0.026ppb |
|---|---|
| HF/VHF VFO Amateur & HF Radio | Temp = -40 - +85 degC |
| Clock Data Communications | Differential or CMOS output |
| **Si514 Important Features** | 5x7mm or 3.2x5mm packages |
| Choice of Vcc = 1.8, 2.5 or 3.3VDC | Min. Ext. Comps = 4xR + 1xC |
| Frequency Range 100KHz - 125/250MHz | Cost Approx $16.00 Cdn (Digikey) |



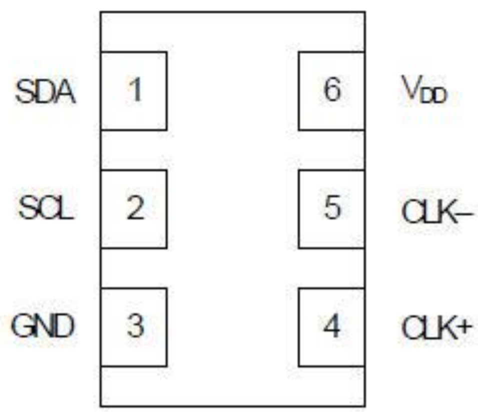
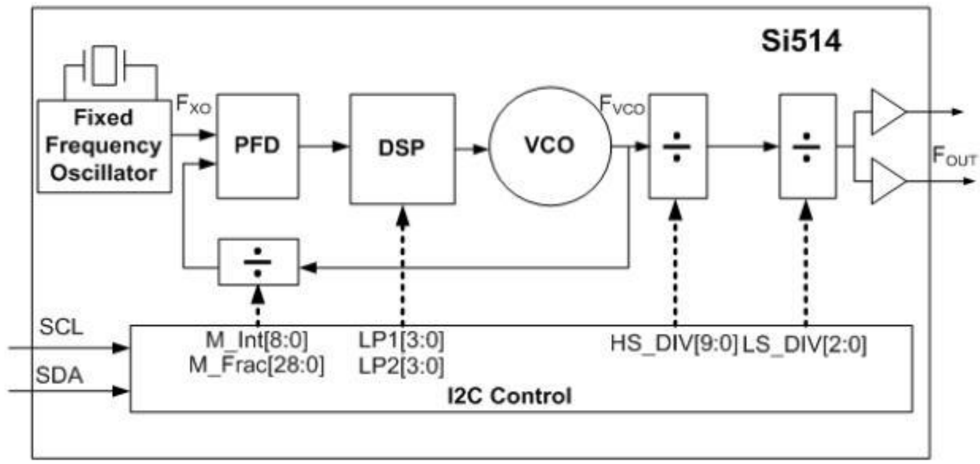




**Figure 1.1 Si514 Typical PCB Mounting on** sdr_uc & **Pinout**

## 1.2 - Si514 Block Diagram




$$F_{out} = \frac{F_{XO} \times M}{HS\_DIV \times LS\_DIV}$$

where $F_{XO} = 31.98\text{MHz}$

**Figure 1.2 Si514 Block Diagram & Programming Equation**

Figure 1.2 shows the Si514 block diagram and programming equation. The block diagram shows a common frequency synthesizer structure consisting of a VCO, feedback dividers, phase/frequency detector and frequency reference. Frequency synthesizers are discussed in detail in Ref.3 Chapter 6. The Frequency Output Fout is obtained by setting the Feedback Multiplier M=M_Int.M_Frac, High Speed Divider HS_DIV and Low Speed Divider LS_DIV. The following parameters apply:

- Fxo = 31.98MHz
- M = High resolution 29bit fractional multiplier
- 65.04065041 =< M <= 78.17385866
- 2080 MHz =< FVCO <= 2500 MHz
- 2080MHz = 31.98MHz * 65.04065041
- 2500MHz = 31.98MHz * 78.17385866
- Fout startup = 10.0MHz (depends on ordering code)
- Small Frequency Change < +/- 1000ppm Fout, action keep Fvco & change M
- Large Frequency Change >= +/- 1000ppm Fout, change Fvco, change HS_DIV, LS_DIV, LP1, LP2 and recalibration

There are two basic situations for frequency change. When the Si514 starts it assumes the factory programmed frequency, in this case 10MHz. When a new frequency is desired, the ppm change required in Fout needs to be determined. If it is less than 1000ppm, then this is the small frequency change case. In the small frequency case, Fvco stays the same, only M is changed. Fout changes continuously to the final value in approx. 100usec. If the change is >= 1000ppm, then this is the large frequency change case. Fvco changes along with M, HS_DIV, LS_DIV, LP1 and LP2.

## 1.3 - Case 1 Small Frequency Change < +/- 1000ppm

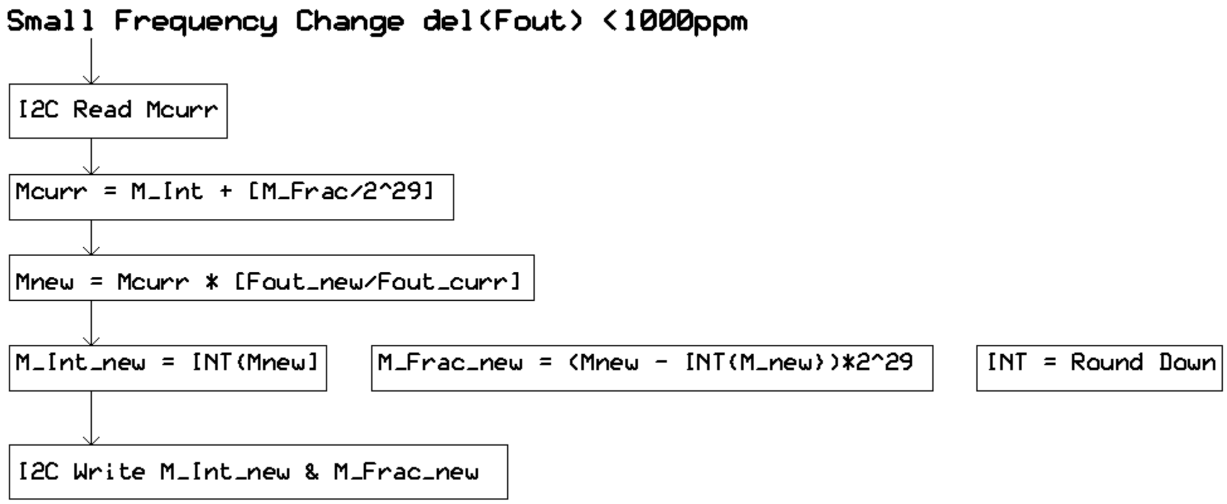



**Figure 1.3 Si514 Programming Small Frequency Change < +/-1000ppm**

Figure 1.3 shows the logic diagram followed for the case of a small frequency change. Consider the following example. The Si514 starts at 10MHz, the registers are read, then a frequency change to 10.005MHz is required.

- Fout_curr = 10.0MHz, Fout_new = 10.005MHz
- ppm = (+0.005/10.0)*10^6 = +500
- 500 < 1000, so this is the small frequency change case
- Mnew = Mcurr*(Fout_new/Fout_curr)

Figure 1.4 shows the register structure of the Si514. Note that Register 8 is split between M_Int & M_Frac. Figure 1.5 shows a read of all the registers for the start frequency of 10MHz. Figure 1.6 shows the registers for the new frequency of 10.005MHz. The registers are read using Ref.2. Later on, software routines will be developed to read the registers and program the unit. The current value of M is read in as:

- M_Int = Reg9[5,0] + Reg8[7,5] = 9bits
- M_Frac = Reg8[4,0] + Reg7[7,0] + Reg6[7,0] + Reg5[7,0] = 29bits
- Reg5 = $14, Reg6 = $02, Reg7 = $4D, Reg8 = $21, Reg9 = $08
- M_Int = b001000001 = $41 = 65
- M_Frac = $014D0214 = 21,824,020
- M = M_Int +[ M_Frac/2^29] = 65.0406504049897194
- Mnew = 65.0406504049897194 * 10.005/10.0 = 65.0731707301922171
- M_Int_new = 65 = $41
- M_Frac_new = .0731707301922171x2^29 = 39,283,236 (round down)
- M_Frac_new = $2576A24
- Reg5new = $25

- Reg6new = $6A
- Reg7new = $57
- Reg8new = $22
- Reg9new = $08

<table>
<tr><th rowspan="2">Address</th><th colspan="8">Bit</th></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td colspan="4">LP1[3:0]</td><td colspan="4">LP2[3:0]</td></tr>
<tr><td>5</td><td colspan="8">M_Frac [7:0]</td></tr>
<tr><td>6</td><td colspan="8">M_Frac [15:8]</td></tr>
<tr><td>7</td><td colspan="8">M_Frac [23:16]</td></tr>
<tr><td>8</td><td colspan="3">M_Int [2:0]</td><td colspan="5">M_Frac [28:24]</td></tr>
<tr><td>9</td><td></td><td></td><td colspan="6">M_Int [8:3]</td></tr>
<tr><td>10</td><td colspan="8">HS_DIV [7:0]</td></tr>
<tr><td>11</td><td></td><td colspan="3">LS_DIV [ 2:0]</td><td></td><td></td><td colspan="2">HS_DIV [9:8]</td></tr>
<tr><td>14</td><td></td><td></td><td colspan="2">OE_STATE [1:0]</td><td></td><td></td><td></td><td></td></tr>
<tr><td>128</td><td>RST</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>132</td><td></td><td></td><td></td><td></td><td></td><td>OE</td><td></td><td>FCAL</td></tr>
</table>

**Figure 1.4 Si514 Register Map**

Register Viewer - 0000161da386 - I2C address: 0x55

File Help

| addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 22h | 16 | D8h | 32 | 14h | 48 | 9h | 64 | 0h | 80 | 0h | 96 | 0h | 112 | 0h | 128 | 0h | 144 | 0h | 160 | 8h | 176 | 4Fh | 192 | 0h | 208 | 0h | 224 | 0h | 240 | 0h |
| 1 | 4h | 17 | 41h | 33 | 7h | 49 | 0h | 65 | 0h | 81 | 0h | 97 | 0h | 113 | 0h | 129 | F0h | 145 | 8h | 161 | 9Bh | 177 | 21h | 193 | 0h | 209 | 0h | 225 | 0h | 241 | 0h |
| 2 | EAh | 18 | E2h | 34 | Fh | 50 | 83h | 66 | 0h | 82 | 0h | 98 | 0h | 114 | 0h | 130 | 80h | 146 | 50h | 162 | 22h | 178 | 8h | 194 | 0h | 210 | 0h | 226 | 0h | 242 | 0h |
| 3 | Fh | 19 | 0h | 35 | BAh | 51 | F2h | 67 | 0h | 83 | 0h | 99 | 0h | 115 | 0h | 131 | 2h | 147 | 0h | 163 | 0h | 179 | 0h | 195 | 0h | 211 | 0h | 227 | 0h | 243 | 0h |
| 4 | 48h | 20 | 0h | 36 | 77h | 52 | 0h | 68 | 0h | 84 | 0h | 100 | 0h | 116 | 0h | 132 | 4h | 148 | 48h | 164 | 40h | 180 | 0h | 196 | 0h | 212 | 0h | 228 | 0h | 244 | 0h |
| 5 | 14h | 21 | 0h | 37 | 77h | 53 | 0h | 69 | 0h | 85 | 0h | 101 | 0h | 117 | 0h | 133 | EAh | 149 | 3h | 165 | Fh | 181 | 0h | 197 | 0h | 213 | 0h | 229 | 0h | 245 | 0h |
| 6 | 2h | 22 | 0h | 38 | A5h | 54 | 0h | 70 | 0h | 86 | 0h | 102 | 0h | 118 | 0h | 134 | Fh | 150 | 0h | 166 | 0h | 182 | 0h | 198 | 0h | 214 | 0h | 230 | 0h | 246 | 0h |
| 7 | 4Dh | 23 | 0h | 39 | BBh | 55 | 0h | 71 | 0h | 87 | 0h | 103 | 0h | 119 | 0h | 135 | 0h | 151 | 0h | 167 | 10h | 183 | 0h | 199 | 0h | 215 | 0h | 231 | 0h | 247 | 0h |
| 8 | 21h | 24 | 4h | 40 | 8Dh | 56 | 0h | 72 | 0h | 88 | 0h | 104 | 0h | 120 | 0h | 136 | 0h | 152 | 0h | 168 | 0h | 184 | 0h | 200 | 0h | 216 | 0h | 232 | 0h | 248 | 0h |
| 9 | 8h | 25 | 2Fh | 41 | Fh | 57 | 0h | 73 | 0h | 89 | 0h | 105 | 0h | 121 | 0h | 137 | 0h | 153 | 0h | 169 | 0h | 185 | 0h | 201 | 0h | 217 | 0h | 233 | 0h | 249 | 0h |
| 10 | D0h | 26 | 0h | 42 | 8Bh | 58 | 0h | 74 | 0h | 90 | 0h | 106 | 0h | 122 | 0h | 138 | 0h | 154 | 0h | 170 | 50h | 186 | 0h | 202 | 0h | 218 | 0h | 234 | 0h | 250 | 0h |
| 11 | 0h | 27 | C0h | 43 | 8h | 59 | 0h | 75 | 0h | 91 | 0h | 107 | 0h | 123 | 0h | 139 | ABh | 155 | 0h | 171 | 12h | 187 | 0h | 203 | 0h | 219 | 0h | 235 | 0h | 251 | 0h |
| 12 | 0h | 28 | 31h | 44 | 0h | 60 | 0h | 76 | 0h | 92 | 0h | 108 | 0h | 124 | 0h | 140 | 0h | 156 | 0h | 172 | 0h | 188 | 0h | 204 | 0h | 220 | 0h | 236 | 0h | 252 | 0h |
| 13 | 0h | 29 | 2h | 45 | 60h | 61 | 0h | 77 | 0h | 93 | 0h | 109 | 0h | 125 | 0h | 141 | 0h | 157 | 0h | 173 | 0h | 189 | 0h | 205 | 0h | 221 | 0h | 237 | 0h | 253 | 0h |
| 14 | 80h | 30 | Ah | 46 | 0h | 62 | 0h | 78 | 0h | 94 | 0h | 110 | 0h | 126 | 0h | 142 | 0h | 158 | 0h | 174 | B7h | 190 | 0h | 206 | 0h | 222 | 0h | 238 | 0h | 254 | 0h |
| 15 | 55h | 31 | 3Dh | 47 | 0h | 63 | 0h | 79 | 0h | 95 | 0h | 111 | 0h | 127 | 0h | 143 | 0h | 159 | 42h | 175 | 2Dh | 191 | 0h | 207 | 0h | 223 | 0h | 239 | 0h | 255 | 0h |

**Figure 1.5 Si514 Register Read Start Frequency = 10MHz**

Register Viewer - 0000161da386 - I2C address: 0x55

File Help

| addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 22h | 16 | D8h | 32 | 14h | 48 | 9h | 64 | 0h | 80 | 0h | 96 | 0h | 112 | 0h |
| 1 | 4h | 17 | 41h | 33 | 7h | 49 | 0h | 65 | 0h | 81 | 0h | 97 | 0h | 113 | 0h |
| 2 | EAh | 18 | E2h | 34 | Fh | 50 | 83h | 66 | 0h | 82 | 0h | 98 | 0h | 114 | 0h |
| 3 | Fh | 19 | 0h | 35 | BAh | 51 | F2h | 67 | 0h | 83 | 0h | 99 | 0h | 115 | 0h |
| 4 | 48h | 20 | 0h | 36 | 77h | 52 | 0h | 68 | 0h | 84 | 0h | 100 | 0h | 116 | 0h |
| 5 | 25h | 21 | 0h | 37 | 77h | 53 | 0h | 69 | 0h | 85 | 0h | 101 | 0h | 117 | 0h |
| 6 | 6Ah | 22 | 0h | 38 | A5h | 54 | 0h | 70 | 0h | 86 | 0h | 102 | 0h | 118 | 0h |
| 7 | 57h | 23 | 0h | 39 | BBh | 55 | 0h | 71 | 0h | 87 | 0h | 103 | 0h | 119 | 0h |
| 8 | 22h | 24 | 4h | 40 | 8Dh | 56 | 0h | 72 | 0h | 88 | 0h | 104 | 0h | 120 | 0h |
| 9 | 8h | 25 | 2Fh | 41 | Fh | 57 | 0h | 73 | 0h | 89 | 0h | 105 | 0h | 121 | 0h |
| 10 | D0h | 26 | 0h | 42 | 8Bh | 58 | 0h | 74 | 0h | 90 | 0h | 106 | 0h | 122 | 0h |
| 11 | 0h | 27 | C0h | 43 | 8h | 59 | 0h | 75 | 0h | 91 | 0h | 107 | 0h | 123 | 0h |
| 12 | 0h | 28 | 31h | 44 | 0h | 60 | 0h | 76 | 0h | 92 | 0h | 108 | 0h | 124 | 0h |
| 13 | 0h | 29 | 2h | 45 | 60h | 61 | 0h | 77 | 0h | 93 | 0h | 109 | 0h | 125 | 0h |
| 14 | 80h | 30 | Ah | 46 | 0h | 62 | 0h | 78 | 0h | 94 | 0h | 110 | 0h | 126 | 0h |
| 15 | 55h | 31 | 3Dh | 47 | 0h | 63 | 0h | 79 | 0h | 95 | 0h | 111 | 0h | 127 | 0h |

| addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 128 | 0h | 144 | 0h | 160 | 28h | 176 | 59h | 192 | 0h | 208 | 0h | 224 | 0h | 240 | 0h |
| 129 | F0h | 145 | 8h | 161 | 86h | 177 | 22h | 193 | 0h | 209 | 0h | 225 | 0h | 241 | 0h |
| 130 | 80h | 146 | 50h | 162 | 32h | 178 | 8h | 194 | 0h | 210 | 0h | 226 | 0h | 242 | 0h |
| 131 | 2h | 147 | 0h | 163 | 0h | 179 | 0h | 195 | 0h | 211 | 0h | 227 | 0h | 243 | 0h |
| 132 | 4h | 148 | 48h | 164 | 40h | 180 | 0h | 196 | 0h | 212 | 0h | 228 | 0h | 244 | 0h |
| 133 | EAh | 149 | 3h | 165 | Fh | 181 | 0h | 197 | 0h | 213 | 0h | 229 | 0h | 245 | 0h |
| 134 | Fh | 150 | 0h | 166 | 0h | 182 | 0h | 198 | 0h | 214 | 0h | 230 | 0h | 246 | 0h |
| 135 | 0h | 151 | 0h | 167 | 10h | 183 | 0h | 199 | 0h | 215 | 0h | 231 | 0h | 247 | 0h |
| 136 | 0h | 152 | 0h | 168 | 0h | 184 | 0h | 200 | 0h | 216 | 0h | 232 | 0h | 248 | 0h |
| 137 | 0h | 153 | 0h | 169 | 0h | 185 | 0h | 201 | 0h | 217 | 0h | 233 | 0h | 249 | 0h |
| 138 | 0h | 154 | 0h | 170 | 50h | 186 | 0h | 202 | 0h | 218 | 0h | 234 | 0h | 250 | 0h |
| 139 | ABh | 155 | 0h | 171 | 12h | 187 | 0h | 203 | 0h | 219 | 0h | 235 | 0h | 251 | 0h |
| 140 | 0h | 156 | 0h | 172 | 0h | 188 | 0h | 204 | 0h | 220 | 0h | 236 | 0h | 252 | 0h |
| 141 | 0h | 157 | 0h | 173 | 0h | 189 | 0h | 205 | 0h | 221 | 0h | 237 | 0h | 253 | 0h |
| 142 | 0h | 158 | 0h | 174 | 10h | 190 | 0h | 206 | 0h | 222 | 0h | 238 | 0h | 254 | 0h |
| 143 | 0h | 159 | 97h | 175 | 96h | 191 | 0h | 207 | 0h | 223 | 0h | 239 | 0h | 255 | 0h |

Data Format
Decimal
Hex

Read All

**Figure 1.6 Si514 Register Read New Frequency = 10.005MHz**

Scilab program si514sf.sce can be used to calculate all the values for "Case 1 Small Frequency Change". Scilab (Ref.4) is an Open Source C like program for scientific computation. The complete program is listed in Appendix A.

si514sf.sce

```
//Si514 Programming 514CBB000112AAG
//Small Frequency Change Case < +/-1000ppm
//Default Frequency = 10MHz
//Frequency in MHz
//J.Clark April 24th - 2016

//Startup Conditions for Fout=10MHz
//FOC=10MHz
//Fxo=31.98MHz
//Read Registers EVB Tool
FOC=10.0;
R5h='14';
R6h='02';
R7h='4D';
R8h='21';
R9h='08';
R75h=R7h+R6h+R5h;
R5=uint8(hex2dec(R5h));
R6=uint8(hex2dec(R6h));
R7=uint8(hex2dec(R7h));
R8=uint8(hex2dec(R8h));
R9=uint8(hex2dec(R9h));
R75=hex2dec(R75h);

//Calculate MC = M_IC + [M_FC/2^29]
M_IC=double(0);
M_FC=double(0);
for j=1:24
    if (bitget(R75,j)==1) then
        M_FC=bitset(M_FC,j);
    end
end
```

Scilab 5.5.2 Console

```
-->R5Nh
 R5Nh  =

 25

-->R6Nh
 R6Nh  =

 6A

-->R7Nh
 R7Nh  =

 57

-->R8Nh
 R8Nh  =

 22

-->R9Nh
 R9Nh  =

 8
```

**Figure 1.7 Si514 Program for Case 1 Small Frequency change** si514sf.sce

## 1.4 - Case 2 Large Frequency Change >= +/- 1000ppm

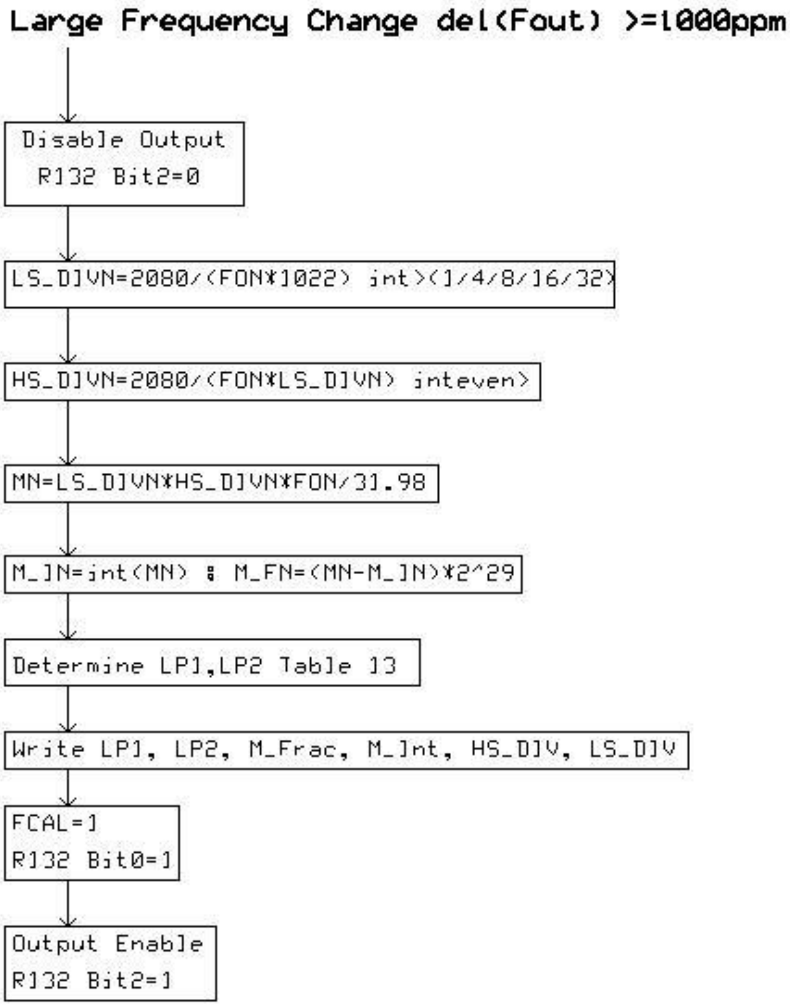



**Figure 1.8 Si514 Programming Large Frequency Change Case >= +/-1000ppm**

Figure 1.8 shows the logic diagram followed for the case of a large frequency change. The output is first turned off by setting the OE bit R132 bit 2=0. The dividers LS_DIV, HS_DIV are calculated according to formulas and then M, M_Int and M_Frac are calculated according to formulas. Once they are calculated, the registers are written in the order indicated, the calibration FCAL=1 is set and then the OE=1 output enabled.

Consider an example. The startup frequency is 10MHz. Let's assume a new frequency of 125MHz is desired. Scilab program si514lf.sce can be used to determine the various parameters. The program is listed in Appendix B. Note that Scilab uses double precision for the variables, so the results may be slightly different than the data sheet depending on the example. Ref.5 discusses using ScicosLab/Scilab. The program gives the following results:

- Frequency change = [(125-10)/10]*10^6 = 11500000 >> 1000, Case 2
- LS_DIVN = 1, so output set to 0 for bypass, LS_DIV = 0
- HS_DIVN = 18 = $12
- MN = 70.3564727954971829
- M_IN = 70 = $46, M_FN = 191379875 = $B6839A3
- LP1 = 3, LP2 =3
- Reg0 = $33
- Reg5 = $A3, Reg6 = $39, Reg7 = $68
- Reg8 = $CB, Reg9 = $08, Reg10 = $12, R11 = $00

si514lf.sce

```
//Si514 Programming 514CBB000112AAG
//Large Frequency Change Case >= +/-1000ppm
//Frequencies in MHz
//Speed Grade Reg48 & Reg49
//J.Clark April 2nd - 2016

//Enter New Frequency
//Fxo=31.98MHz
FON=125;
LS_DIVN=2080/(FON*1022);
if (LS_DIVN<1.0) then
    LS_DIVN=1;
elseif ((LS_DIVN>1.0) & (LS_DIVN< 2.0)) then
    LS_DIVN=2;
elseif ((LS_DIVN>2.0) & (LS_DIVN< 4.0)) then
    LS_DIVN=4;
elseif ((LS_DIVN>4.0) & (LS_DIVN< 8.0)) then
    LS_DIVN=8;
elseif ((LS_DIVN>8.0) & (LS_DIVN< 16.0)) then
    LS_DIVN=16;
else (LS_DIVN>16.0)
    LS_DIVN=32;
end
HS_DIVN=ceil(2080/(FON*LS_DIVN));
if (modulo(HS_DIVN,2)==0) then
    HS_DIVN=HS_DIVN
else
    HS_DIVN=HS_DIVN+1;
end
MN=LS_DIVN*HS_DIVN*(FON/31.98);
M_IN=int(MN);
```

Scilab 5.5.2 Console

```
-->exec('W:\si514\r

-->LP1
 LP1  =

    3.

-->LP2
 LP2  =

    3.

-->M_FN
 M_FN  =

    191379875.

-->M_IN
 M_IN  =

    70.

-->HS_DIVN
 HS_DIVN  =

    18.

-->LS_DIVN
 LS_DIVN  =

    1.
```

**Figure 1.9 Si514 Case 2 FON = 125MHz Large Frequency change** si514lf.sce

Register Viewer - 0000161da386 - I2C address: 0x55

File Help

| addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 33h | 16 | D8h | 32 | 14h | 48 | 9h | 64 | 0h | 80 | 0h | 96 | 0h | 112 | 0h | 128 | 0h | 144 | 0h | 160 | 3Ch | 176 | 6Ah | 192 | 0h | 208 | 0h | 224 | 0h | 240 | 0h |
| 1 | 4h | 17 | 41h | 33 | 7h | 49 | 0h | 65 | 0h | 81 | 0h | 97 | 0h | 113 | 0h | 129 | F0h | 145 | 8h | 161 | 26h | 177 | CBh | 193 | 0h | 209 | 0h | 225 | 0h | 241 | 0h |
| 2 | EAh | 18 | E2h | 34 | Fh | 50 | 83h | 66 | 0h | 82 | 0h | 98 | 0h | 114 | 0h | 130 | 80h | 146 | 50h | 162 | 14h | 178 | 8h | 194 | 0h | 210 | 0h | 226 | 0h | 242 | 0h |
| 3 | Fh | 19 | 0h | 35 | BAh | 51 | F2h | 67 | 0h | 83 | 0h | 99 | 0h | 115 | 0h | 131 | 2h | 147 | 0h | 163 | 0h | 179 | 0h | 195 | 0h | 211 | 0h | 227 | 0h | 243 | 0h |
| 4 | 48h | 20 | 0h | 36 | 77h | 52 | 0h | 68 | 0h | 84 | 0h | 100 | 0h | 116 | 0h | 132 | 4h | 148 | 48h | 164 | 40h | 180 | 0h | 196 | 0h | 212 | 0h | 228 | 0h | 244 | 0h |
| 5 | A2h | 21 | 0h | 37 | 77h | 53 | 0h | 69 | 0h | 85 | 0h | 101 | 0h | 117 | 0h | 133 | C5h | 149 | 3h | 165 | Fh | 181 | 0h | 197 | 0h | 213 | 0h | 229 | 0h | 245 | 0h |
| 6 | 39h | 22 | 0h | 38 | A5h | 54 | 0h | 70 | 0h | 86 | 0h | 102 | 0h | 118 | 0h | 134 | 3h | 150 | 0h | 166 | 0h | 182 | 0h | 198 | 0h | 214 | 0h | 230 | 0h | 246 | 0h |
| 7 | 68h | 23 | 0h | 39 | BBh | 55 | 0h | 71 | 0h | 87 | 0h | 103 | 0h | 119 | 0h | 135 | 0h | 151 | 0h | 167 | 10h | 183 | 0h | 199 | 0h | 215 | 0h | 231 | 0h | 247 | 0h |
| 8 | CBh | 24 | 4h | 40 | 8Dh | 56 | 0h | 72 | 0h | 88 | 0h | 104 | 0h | 120 | 0h | 136 | 0h | 152 | 0h | 168 | 0h | 184 | 0h | 200 | 0h | 216 | 0h | 232 | 0h | 248 | 0h |
| 9 | 8h | 25 | 2Fh | 41 | Fh | 57 | 0h | 73 | 0h | 89 | 0h | 105 | 0h | 121 | 0h | 137 | 0h | 153 | 0h | 169 | 0h | 185 | 0h | 201 | 0h | 217 | 0h | 233 | 0h | 249 | 0h |
| 10 | 12h | 26 | 0h | 42 | 8Bh | 58 | 0h | 74 | 0h | 90 | 0h | 106 | 0h | 122 | 0h | 138 | 0h | 154 | 0h | 170 | 50h | 186 | 0h | 202 | 0h | 218 | 0h | 234 | 0h | 250 | 0h |
| 11 | 0h | 27 | C0h | 43 | 8h | 59 | 0h | 75 | 0h | 91 | 0h | 107 | 0h | 123 | 0h | 139 | ABh | 155 | 0h | 171 | 12h | 187 | 0h | 203 | 0h | 219 | 0h | 235 | 0h | 251 | 0h |
| 12 | 0h | 28 | 31h | 44 | 0h | 60 | 0h | 76 | 0h | 92 | 0h | 108 | 0h | 124 | 0h | 140 | 0h | 156 | 0h | 172 | 0h | 188 | 0h | 204 | 0h | 220 | 0h | 236 | 0h | 252 | 0h |
| 13 | 0h | 29 | 2h | 45 | 60h | 61 | 0h | 77 | 0h | 93 | 0h | 109 | 0h | 125 | 0h | 141 | 0h | 157 | 0h | 173 | 0h | 189 | 0h | 205 | 0h | 221 | 0h | 237 | 0h | 253 | 0h |
| 14 | 80h | 30 | Ah | 46 | 0h | 62 | 0h | 78 | 0h | 94 | 0h | 110 | 0h | 126 | 0h | 142 | 0h | 158 | 0h | 174 | AFh | 190 | 0h | 206 | 0h | 222 | 0h | 238 | 0h | 254 | 0h |
| 15 | 55h | 31 | 3Dh | 47 | 0h | 63 | 0h | 79 | 0h | 95 | 0h | 111 | 0h | 127 | 0h | 143 | 0h | 159 | 42h | 175 | 92h | 191 | 0h | 207 | 0h | 223 | 0h | 239 | 0h | 255 | 0h |

**Figure 1.10 Si514 Register Read New Frequency = 125.0MHz**

Consider a second example. The existing frequency is 125MHz. The new frequency is now 25MHz. The program gives the following results:

- Frequency change = [(25-125)/125]*10^6 = -- 800000 << --1000, Case 2
- LS_DIVN = 1, so output set to 0 for bypass, LS_DIV = 0
- HS_DIVN = 84 = $54
- MN = 65.6660412757973688
- M_IN = 65 = $41, M_FN = 357578187  = $ 155035CB
- LP1 = 2, LP2 =3
- Reg0 = $23
- Reg5 = $CB, Reg6 = $35, Reg7 = $50
- Reg8  = $35, Reg9 = $08, Reg10 = $54, R11 = $00

Register Viewer - 0000161da386 - I2C address: 0x55

File Help

| addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data | addr | data |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 23h | 16 | D8h | 32 | 14h | 48 | 9h | 64 | 0h | 80 | 0h | 96 | 0h | 112 | 0h | 128 | 0h | 144 | 0h | 160 | 0h | 176 | 52h | 192 | 0h | 208 | 0h | 224 | 0h | 240 | 0h |
| 1 | 4h | 17 | 41h | 33 | 7h | 49 | 0h | 65 | 0h | 81 | 0h | 97 | 0h | 113 | 0h | 129 | F0h | 145 | 8h | 161 | CBh | 177 | 35h | 193 | 0h | 209 | 0h | 225 | 0h | 241 | 0h |
| 2 | EAh | 18 | E2h | 34 | Fh | 50 | 83h | 66 | 0h | 82 | 0h | 98 | 0h | 114 | 0h | 130 | 80h | 146 | 50h | 162 | 16h | 178 | 8h | 194 | 0h | 210 | 0h | 226 | 0h | 242 | 0h |
| 3 | Fh | 19 | 0h | 35 | BAh | 51 | F2h | 67 | 0h | 83 | 0h | 99 | 0h | 115 | 0h | 131 | 2h | 147 | 0h | 163 | 0h | 179 | 0h | 195 | 0h | 211 | 0h | 227 | 0h | 243 | 0h |
| 4 | 48h | 20 | 0h | 36 | 77h | 52 | 0h | 68 | 0h | 84 | 0h | 100 | 0h | 116 | 0h | 132 | 4h | 148 | 48h | 164 | 40h | 180 | 0h | 196 | 0h | 212 | 0h | 228 | 0h | 244 | 0h |
| 5 | CBh | 21 | 0h | 37 | 77h | 53 | 0h | 69 | 0h | 85 | 0h | 101 | 0h | 117 | 0h | 133 | D4h | 149 | 3h | 165 | Fh | 181 | 0h | 197 | 0h | 213 | 0h | 229 | 0h | 245 | 0h |
| 6 | 35h | 22 | 0h | 38 | A5h | 54 | 0h | 70 | 0h | 86 | 0h | 102 | 0h | 118 | 0h | 134 | Fh | 150 | 0h | 166 | 0h | 182 | 0h | 198 | 0h | 214 | 0h | 230 | 0h | 246 | 0h |
| 7 | 50h | 23 | 0h | 39 | BBh | 55 | 0h | 71 | 0h | 87 | 0h | 103 | 0h | 119 | 0h | 135 | 0h | 151 | 0h | 167 | 10h | 183 | 0h | 199 | 0h | 215 | 0h | 231 | 0h | 247 | 0h |
| 8 | 35h | 24 | 4h | 40 | 8Dh | 56 | 0h | 72 | 0h | 88 | 0h | 104 | 0h | 120 | 0h | 136 | 0h | 152 | 0h | 168 | 0h | 184 | 0h | 200 | 0h | 216 | 0h | 232 | 0h | 248 | 0h |
| 9 | 8h | 25 | 2Fh | 41 | Fh | 57 | 0h | 73 | 0h | 89 | 0h | 105 | 0h | 121 | 0h | 137 | 0h | 153 | 0h | 169 | 0h | 185 | 0h | 201 | 0h | 217 | 0h | 233 | 0h | 249 | 0h |
| 10 | 54h | 26 | 0h | 42 | 8Bh | 58 | 0h | 74 | 0h | 90 | 0h | 106 | 0h | 122 | 0h | 138 | 0h | 154 | 0h | 170 | 50h | 186 | 0h | 202 | 0h | 218 | 0h | 234 | 0h | 250 | 0h |
| 11 | 0h | 27 | C0h | 43 | 8h | 59 | 0h | 75 | 0h | 91 | 0h | 107 | 0h | 123 | 0h | 139 | ABh | 155 | 0h | 171 | 12h | 187 | 0h | 203 | 0h | 219 | 0h | 235 | 0h | 251 | 0h |
| 12 | 0h | 28 | 31h | 44 | 0h | 60 | 0h | 76 | 0h | 92 | 0h | 108 | 0h | 124 | 0h | 140 | 0h | 156 | 0h | 172 | 0h | 188 | 0h | 204 | 0h | 220 | 0h | 236 | 0h | 252 | 0h |
| 13 | 0h | 29 | 2h | 45 | 60h | 61 | 0h | 77 | 0h | 93 | 0h | 109 | 0h | 125 | 0h | 141 | 0h | 157 | 0h | 173 | 0h | 189 | 0h | 205 | 0h | 221 | 0h | 237 | 0h | 253 | 0h |
| 14 | 80h | 30 | Ah | 46 | 0h | 62 | 0h | 78 | 0h | 94 | 0h | 110 | 0h | 126 | 0h | 142 | 0h | 158 | 0h | 174 | C6h | 190 | 0h | 206 | 0h | 222 | 0h | 238 | 0h | 254 | 0h |
| 15 | 55h | 31 | 3Dh | 47 | 0h | 63 | 0h | 79 | 0h | 95 | 0h | 111 | 0h | 127 | 0h | 143 | 0h | 159 | F3h | 175 | 66h | 191 | 0h | 207 | 0h | 223 | 0h | 239 | 0h | 255 | 0h |

**Figure 1.11 Si514 Register Read New Frequency = 25.0MHz**

si514lf.sce

```
//Si514 Programming 514CBB000112AAG
//Large Frequency Change Case >= +/-1000ppm
//Frequencies in MHz
//FXO=31.98MHz
//Speed Grade Reg48 & Reg49
//J.Clark April 4th - 2016

//Enter New Frequency FON
//Calculate LS_DIVN, HS_DIVN, MN, M_IN, M_FN, LP1, LP2
//LS_DIVN=1 means bypass set LS_DIVN=0
FXO=31.98;
FON=25.0;
LS_DIVN=2080/(FON*1022);
if (LS_DIVN<1.0) then
    LS_DIVN=1;
elseif ((LS_DIVN>1.0) & (LS_DIVN< 2.0)) then
    LS_DIVN=2;
elseif ((LS_DIVN>2.0) & (LS_DIVN< 4.0)) then
    LS_DIVN=4;
elseif ((LS_DIVN>4.0) & (LS_DIVN< 8.0)) then
    LS_DIVN=8;
elseif ((LS_DIVN>8.0) & (LS_DIVN< 16.0)) then
    LS_DIVN=16;
else (LS_DIVN>16.0)
    LS_DIVN=32;
end
HS_DIVN=ceil(2080/(FON*LS_DIVN));
if (modulo(HS_DIVN,2)==0) then
    HS_DIVN=HS_DIVN
else
    HS_DIVN=HS_DIVN+1;
end
MN=LS_DIVN*HS_DIVN*(FON/FXO);
M_IN=int(MN);
```

Scilab 5.5.2 Console

```
-->HS_DIVN
 HS_DIVN  =

    84.

-->LS_DIVN
 LS_DIVN  =

    0.

-->MN
 MN  =

    65.666041275797368

-->M_IN
 M_IN  =

    65.

-->M_FN
 M_FN  =

    357578187.

-->LP1
 LP1  =

    2.

-->LP2
 LP2  =

    3.
```

**Figure 1.12 Si514 Case 2 FON = 25MHz Large Frequency change** si514lf.sce

## 1.5 - Si514 I2C Read Word Protocol

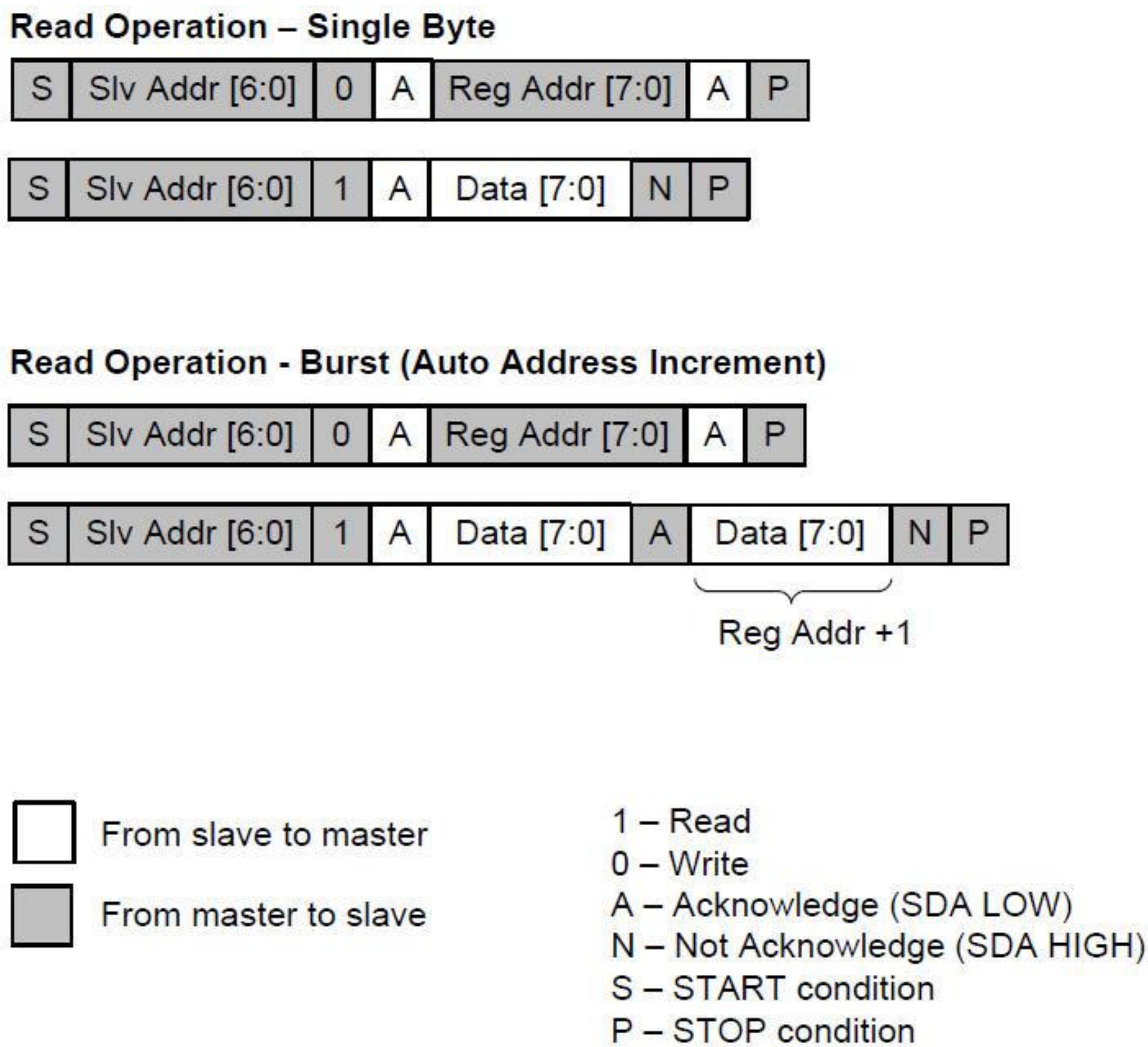



**Figure 1.10 Si514 I2C Read Protocol**

The Si514 works in 7bit address slave mode at normal 100Kbps or fast 400kbps speed. The part used in this document has address $55 = 1010101 = 7bits. Figure 1.10 shows the protocol for a single register read. First a Start bit is sent, followed by the 7bit Si514 address. Then a write 0 is sent, the Si514 ACKs (pulls low), the 8bit Register address is sent, ACK'd, finally a stop bit is sent. Next another Start bit, followed by the Si514 7bit address again, a read 1 bit, ACK by the Si514, and the data from the Register addressed in the previous packet, NAK (set high) by the Si514, followed by a Stop. Note that a single Register Read is a two packet process.

## 1.6 - Si514 I2C Write Word Protocol

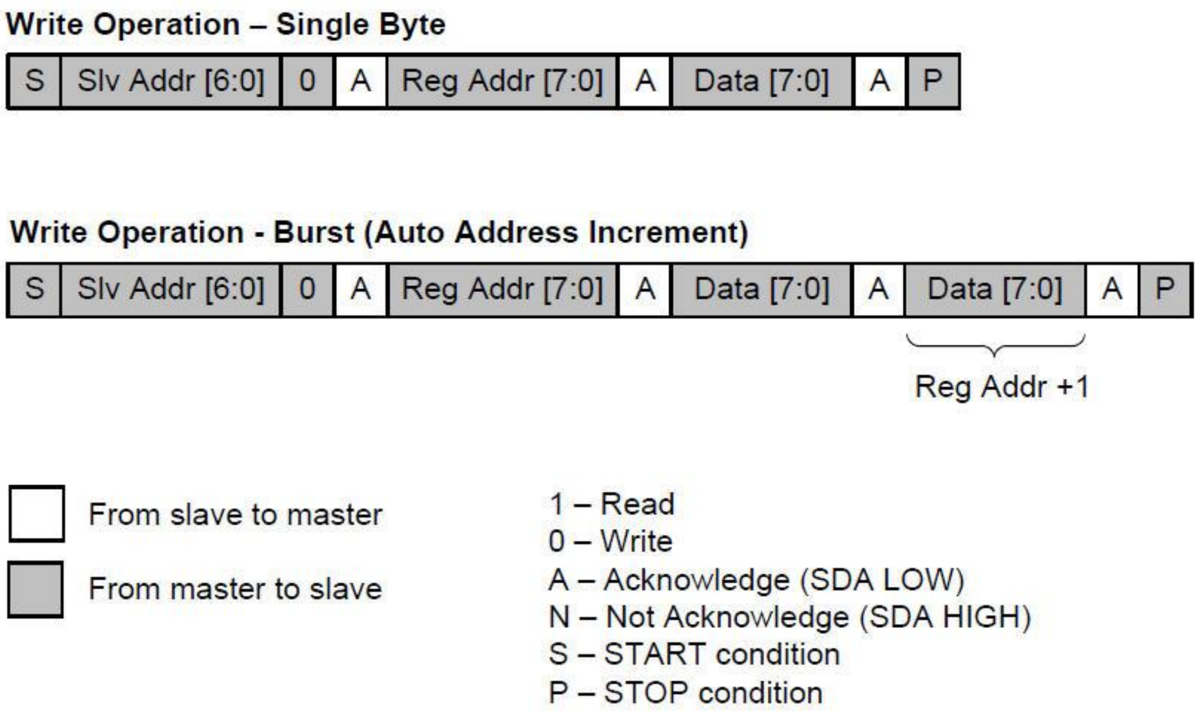



**Figure 1.11 Si514 I2C Write Protocol**

Figure 1.11 shows the Si514 write protocol for a single register write. First a Start bit is sent, followed by the 7bit Si514 slave address ($55). A write bit = 0 is then sent, ACK'd by the Si514 slave, then the 8bit Register address is sent, ACK'd again by the Si514, followed by the data for that Register, ACK'd and finally the Stop bit is sent.

## 1.7 - I2C Specification NXP UM10204

Before discussing how to implement the I2C Read & Write Waveforms of Figures 1.10 & 1.11, let's review some I2C basics from the design specification from Philips Semiconductors now NXP-UM10204 Ref.6. Figure 1.12 shows the START & STOP conditions sent by the Master, as described in the specification.

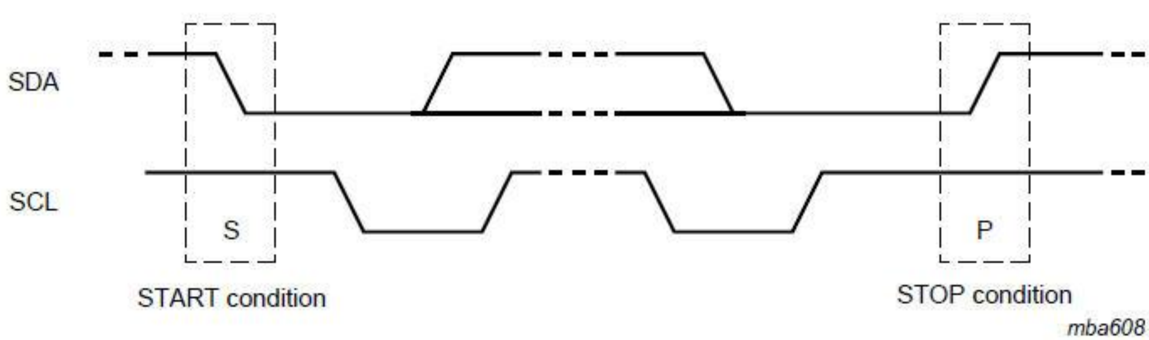



**Figure 1.12 I2C START/STOP Waveforms**

For the START condition, the SDA goes from high to low while the SCL is high. For the STOP condition, the SDA goes from low to high while SCL is high. Figure 1.13 shows the data transfer on the bus between Master and Slave. A START condition is first established, then 1 byte of data is sent MSB first and an ACK is received from the slave. For an ACK, the Master releases the SDA line and during the 9th clock pulse, the slave pulls the SDA low. If the SDA stays high, this means a NAK is received, or the byte was not received correctly. The SCL line is then held low by the Master to service any interrupts. Following this, the next byte is sent, followed by an ACK from the Slave and finally a STOP is sent.

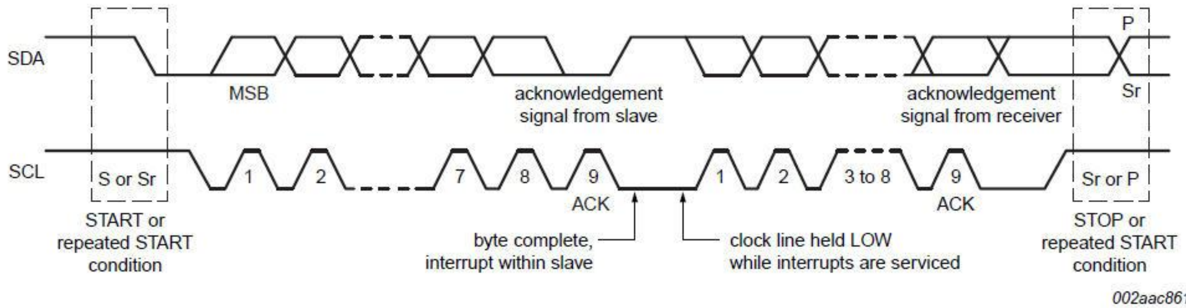



**Figure 1.13 I2C Bus Data Transfer**

## 1.8 - I2C Bus Programming Approach

In the next chapters we will use two different approaches to program the Si514 using popular devices. The first approach will be to use the Arduino Uno Rev.3 (Atmel ATmega328 Microcontroller) using the Arduino IDE. Finally we will use the ever popular Raspberry Pi 2B using the Linux environment with Python3.4.

Note that all code is available in printed form in the Appendices as well as a downloadable zip file.