: Variables:
reg[1] - reg[4] = data
Reg[5] = new data
Reg[6] = calculations
Reg[7] = F3 coefficient
Reg[8] = F2 coefficient
Reg[9] = F1 coefficient
Reg[10] = F0 Coefficient

```
1.  Idle:   if (data_ready=0)  goto idle;          // wait until data_ready=1
2.  store: if (data_ready=0)   goto eidle;
3.  reg[5] = data;                                 // Store data in a register
4.  err = 0; // reset error
5.  zero: reg[0] = 0;                              // zero out accumulator
6.   sort1: reg[1] = reg[2];                       // Reorder registers
7.   sort2: reg[2] = reg[3];                       // Reorder registers
8.   sort3: reg[3] = reg[4];                       // Reorder registers
9.   sort4: reg[4] = reg[5];                       // Reorder registers
10.  mul1: reg[6] = reg[4] * reg[7];               // sample4 * F0
11.  sub: reg[0] = reg[0] - reg[6];                // sub small neg. Coefficient
12.  if (V) goto eidle;                            // On overflow, err condition
13.  mul2: reg[6] = reg[3] * reg[8];               // sample3* F1
14.  add: reg[0] = reg[0] + reg[6];                // add large pos. Coefficient
15.  if (V) goto eidle;                            // On overflow, err condition
16.  mul3: reg[6] = reg[2] * reg[9];               // sample2 * F2
17.  sub: reg[0] = reg[0] - reg[6];                // sub large neg. Coefficient
18.  if (V) goto eidle;                            // On overflow, err condition
19.  mul4: reg[6] = reg[1] * reg[10];              // sample1 * F3
20.  add: reg[0] = reg[0] + reg[6];                // add small pos. Coefficient
21.  if (V) goto eidle;                            // On overflow, err condition
22.  else goto idle;
23.  eidle: err = 1;
24.  if (data_ready=1) goto store;                 // wait until data_ready=1
25.  if (data_ready=0) goto eidle;
```