

Recap

I/O synchronization
 Interrupts
 Output compare periodic interrupts

Overview

Arrays, Table lookup
 Extended math

Proper ways to acknowledge (software makes C3F zero)

<pre>ldaa #\$08 staa TFLG1 movb #\$08, TFLG1</pre>	<pre>TFLG1 = 0x08;</pre>
---	--------------------------

These attempts have no effect on TFLG1

<pre>ldaa #\$00 staa TFLG1 clr TFLG1</pre>	<pre>TFLG1 = 0;</pre>
---	-----------------------

These attempts clear all bits C7F, C6F... C0F in TFLG1

<pre>bset TFLG1, #\$08 movb #\$FF, TFLG1</pre>	<pre>TFLG1 = 0x08; TFLG1 = 0xFF;</pre>
---	--

Multiply instructions

mul unsigned A*B into D
emul unsigned D*Y into 32-bit Y:D
emuls signed D*RegY into RegY:D

Divide instructions

idiv unsigned D/X into X, D remainder
idivs signed D/X into X, D remainder
fdiv unsigned (D:0)/X into X, D remainder
ediv unsigned (Y:D)/X into Y, D remainder
edivs signed (Y:D)/X into Y, D remainder

Example Count (0 to 199) = $\frac{5}{9} \times \text{Angle}(0 \text{ to } 359)$

```
* Count = (5*Angle) / 9
  ldd Angle
  ldy #5
  emul ; (Y:D) = 5*Angle
  ldx #9
  ediv ; Y = (5*Angle) / 9
  sty Count
```

Example Angle (0 to 359) = $\frac{9}{5} \times \text{Count}(0 \text{ to } 199)$

```
* Angle = (9*Count) / 5 ≈ (65536*Count) / 36409
  ldd Count
  ldx #36409
  fdiv
  stx Angle
```

Example Column (0 to 6) = $\frac{7}{256} \times X_{pos}$ (0 to 255)

```
* Column = (7*Xpos) / 256
  ldaa Xpos
  ldab #7
  mul          ;RegD=7*Xpos
  staa Column ;RegA=(7*Xpos) / 256
```

Unsigned 8 to 16-bit promotion instructions

```
; to promote RegB into RegD
  clra
; to promote RegA into RegX
  tfr A,B
  clra
  tfr D,X
```

Signed 8 to 16-bit promotion instructions

```
sex A,D      (same instruction as tfr A,D)
sex B,D      (same instruction as tfr B,D)
sex A,X      (same instruction as tfr A,X)
sex B,X      (same instruction as tfr B,X)
sex A,Y      (same instruction as tfr A,Y)
sex B,Y      (same instruction as tfr B,Y)
```

16 to 8-bit demotion instructions (signed or unsigned)

```
tfr D,A
tfr D,B
tfr X,A
tfr X,B
tfr Y,A
tfr Y,B
```

Example Column (0 to 6) = $\frac{7}{256} \times X_{pos}$ (0 to 255)

```
* Column = (7*Xpos) / 256
  ldab Xpos
  clra      ;promote
  ldy #7
  emul     ; (Y:D)=Y*D
  ldx #256
  ediv     ; (Y:D)/X into Y, D remainder
  tfr Y,A  ;demote
  staa Column
```

Arrays

Used to store data of the same type.

All elements are the same size and are stored together

I be the row index (starts with 0),

n be the number of bytes for each element

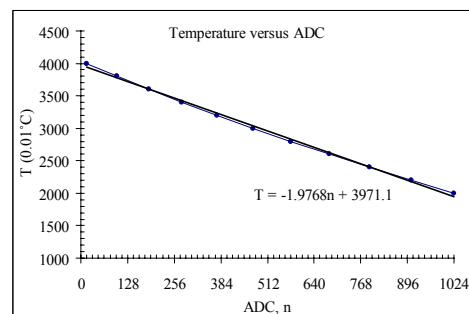
Base is the base address of the array

then the address of the element at **A[I]** is

Base+n*I

Convert ADC data into integer part of fixed point

- 1) Equation
- 2) Little table and interpolation
- 3) Big table and explicit table lookup



1a) Complex Equation, let n be ADC (0 to 1023)

$V = 5.0 * n / 1023$ (in volts)
 $R = 18.31 + 5.312 * V$ (in k Ω)
 $T = 1 / (H_0 + H_1 \ln(R)) - 273.15$ (in $^{\circ}\text{C}$) ($H_0 = 0.002486844$, $H_1 = 0.000243014$)
 $I = 100 * T$ (in 0.01 $^{\circ}\text{C}$)

1b) Simple Equation

$I = 3971 - 1.9768 * n$

Using fdiv, find m such that

$65536 / m = 1.9768$

$m = 65536 / 1.9768 = 33152.5698 \approx 33153$

*** Reg D has ADC result, 0 to 1023**

```

ldx #33153
fdiv
pshx      ; 1.9768*n
ldd #3971
subd 2,SP-
std I     ; 0.01 C
  
```

2) Interpolation

; input from 10-bit ADC (0 to 1023)

```

ADCdata fdb 1023,906,793,683,576,472
        fdb 373,277,186,99,16,0
  
```

; corresponding temperature (0.01 C)

```

Tdata   fdb 2000,2200,2400,2600,2800,3000
        fdb 3200,3400,3600,3800,4000,4000
  
```

Show Help system in TExaS, 6812 programming examples

Search for adjacent points (x1,y1), (x2,y2)

$x1 \leq n < x2$

$I = (n - x1) * (y2 - y1) / (x2 - x1) + y1$

;*****Lookup*****

;Inputs: RegD is 0 to 1023 ADC point, n

; RegD input must be greater than or equal to first Xdata point

; RegD input must be less than last Xdata point

;Output: RegD is I in 0.01 C

;Registers destroyed: X,Y,B,CCR

```

Lookup ldx #ADCdata ; first find x1 <= n < x2
        ldy #Tdata
  
```

```

lookx1 cpd 2,x      ; check n<x2
        blo found  ; stops when X points to x1
        leax 2,x
        leay 2,y
        bra lookx1
  
```

```

found  subd 0,x     ; n-x1
        pshd
        ldd 2,x    ; x2
        subd 0,x   ; D=x2-x1
        tfr D,X   ; X=x2-x1
        puld      ; D=(n-x1)
        fdiv      ; X=(65536*(n-x1))/(x2-x1)
        tfr X,D
        tfr A,B
  
```

; $B = (256 * (n - x1)) / (x2 - x1)$

; Y points to y1,y2

```

        etbl 0,y
  
```

; $D = y1 + B * (y2 - y1)$

```

        rts
  
```

Example: two arrays store calibration data for a thermometer

```

;*****Lookup*****
;Inputs: RegD is 0 to 1023 ADC point, n
;Output: RegD is I in 0.01 C
;Registers destroyed: X,CCR
Lookup ldx #TemperatureTable
        lsl      ; 2*n, 16-bit entries take two bytes each
        ldd     D,X ; TemperatureTable[n]
        rts

TemperatureTable
fdb 4040,4038,4035,4033,4030,4028,4025,4023,4020,4018,4015,4013,4010
fdb 4008,4006,4003,4001,3998,3996,3993,3991,3988,3986,3983,3981,3978
fdb 3976,3973,3971,3969,3966,3964,3961,3959,3956,3954,3951,3949,3947
fdb 3944,3942,3939,3937,3934,3932,3930,3927,3925,3922,3920,3917,3915
fdb 3913,3910,3908,3905,3903,3900,3898,3896,3893,3891,3888,3886,3884
fdb 3881,3879,3876,3874,3872,3869,3867,3864,3862,3860,3857,3855,3853
fdb 3850,3848,3845,3843,3841,3838,3836,3834,3831,3829,3826,3824,3822
fdb 3819,3817,3815,3812,3810,3808,3805,3803,3800,3798,3796,3793,3791
fdb 3789,3786,3784,3782,3779,3777,3775,3772,3770,3768,3765,3763,3761
fdb 3758,3756,3754,3751,3749,3747,3744,3742,3740,3737,3735,3733,3731
fdb 3728,3726,3724,3721,3719,3717,3714,3712,3710,3708,3705,3703,3701
fdb 3698,3696,3694,3691,3689,3687,3685,3682,3680,3678,3675,3673,3671
fdb 3669,3666,3664,3662,3660,3657,3655,3653,3650,3648,3646,3644,3641
fdb 3639,3637,3635,3632,3630,3628,3626,3623,3621,3619,3617,3614,3612
fdb 3610,3608,3605,3603,3601,3599,3596,3594,3592,3590,3587,3585,3583
fdb 3581,3579,3576,3574,3572,3570,3567,3565,3563,3561,3558,3556,3554
fdb 3552,3550,3547,3545,3543,3541,3539,3536,3534,3532,3530,3528,3525
fdb 3523,3521,3519,3517,3514,3512,3510,3508,3506,3503,3501,3499,3497
fdb 3495,3492,3490,3488,3486,3484,3482,3479,3477,3475,3473,3471,3468
fdb 3466,3464,3462,3460,3458,3455,3453,3451,3449,3447,3445,3442,3440
fdb 3438,3436,3434,3432,3430,3427,3425,3423,3421,3419,3417,3414,3412
fdb 3410,3408,3406,3404,3402,3399,3397,3395,3393,3391,3389,3387,3384
fdb 3382,3380,3378,3376,3374,3372,3370,3367,3365,3363,3361,3359,3357
fdb 3355,3353,3350,3348,3346,3344,3342,3340,3338,3336,3334,3331,3329
fdb 3327,3325,3323,3321,3319,3317,3315,3313,3310,3308,3306,3304,3302
fdb 3300,3298,3296,3294,3292,3289,3287,3285,3283,3281,3279,3277,3275
fdb 3273,3271,3269,3267,3264,3262,3260,3258,3256,3254,3252,3250,3248
fdb 3246,3244,3242,3240,3238,3235,3233,3231,3229,3227,3225,3223,3221
fdb 3219,3217,3215,3213,3211,3209,3207,3205,3203,3200,3198,3196,3194
fdb 3192,3190,3188,3186,3184,3182,3180,3178,3176,3174,3172,3170,3168
fdb 3166,3164,3162,3160,3158,3156,3154,3151,3149,3147,3145,3143,3141
fdb 3139,3137,3135,3133,3131,3129,3127,3125,3123,3121,3119,3117,3115
fdb 3113,3111,3109,3107,3105,3103,3101,3099,3097,3095,3093,3091,3089
fdb 3087,3085,3083,3081,3079,3077,3075,3073,3071,3069,3067,3065,3063
fdb 3061,3059,3057,3055,3053,3051,3049,3047,3045,3043,3041,3039,3037
fdb 3035,3033,3031,3029,3027,3025,3023,3021,3019,3017,3015,3013,3011
fdb 3009,3007,3005,3003,3001,2999,2997,2996,2994,2992,2990,2988,2986
fdb 2984,2982,2980,2978,2976,2974,2972,2970,2968,2966,2964,2962,2960
fdb 2958,2956,2954,2952,2950,2948,2947,2945,2943,2941,2939,2937,2935
fdb 2933,2931,2929,2927,2925,2923,2921,2919,2917,2915,2913,2912,2910
fdb 2908,2906,2904,2902,2900,2898,2896,2894,2892,2890,2888,2886,2884
fdb 2883,2881,2879,2877,2875,2873,2871,2869,2867,2865,2863,2861,2859
fdb 2857,2856,2854,2852,2850,2848,2846,2844,2842,2840,2838,2836,2834
fdb 2833,2831,2829,2827,2825,2823,2821,2819,2817,2815,2813,2812,2810
fdb 2808,2806,2804,2802,2800,2798,2796,2794,2793,2791,2789,2787,2785

```

fdb 2783,2781,2779,2777,2776,2774,2772,2770,2768,2766,2764,2762,2760
fdb 2759,2757,2755,2753,2751,2749,2747,2745,2743,2742,2740,2738,2736
fdb 2734,2732,2730,2728,2727,2725,2723,2721,2719,2717,2715,2713,2712
fdb 2710,2708,2706,2704,2702,2700,2698,2697,2695,2693,2691,2689,2687
fdb 2685,2684,2682,2680,2678,2676,2674,2672,2670,2669,2667,2665,2663
fdb 2661,2659,2657,2656,2654,2652,2650,2648,2646,2644,2643,2641,2639
fdb 2637,2635,2633,2632,2630,2628,2626,2624,2622,2620,2619,2617,2615
fdb 2613,2611,2609,2608,2606,2604,2602,2600,2598,2597,2595,2593,2591
fdb 2589,2587,2586,2584,2582,2580,2578,2576,2574,2573,2571,2569,2567
fdb 2565,2564,2562,2560,2558,2556,2554,2553,2551,2549,2547,2545,2543
fdb 2542,2540,2538,2536,2534,2533,2531,2529,2527,2525,2523,2522,2520
fdb 2518,2516,2514,2513,2511,2509,2507,2505,2503,2502,2500,2498,2496
fdb 2494,2493,2491,2489,2487,2485,2484,2482,2480,2478,2476,2474,2473
fdb 2471,2469,2467,2465,2464,2462,2460,2458,2456,2455,2453,2451,2449
fdb 2447,2446,2444,2442,2440,2438,2437,2435,2433,2431,2430,2428,2426
fdb 2424,2422,2421,2419,2417,2415,2413,2412,2410,2408,2406,2404,2403
fdb 2401,2399,2397,2396,2394,2392,2390,2388,2387,2385,2383,2381,2379
fdb 2378,2376,2374,2372,2371,2369,2367,2365,2363,2362,2360,2358,2356
fdb 2355,2353,2351,2349,2347,2346,2344,2342,2340,2339,2337,2335,2333
fdb 2332,2330,2328,2326,2324,2323,2321,2319,2317,2316,2314,2312,2310
fdb 2309,2307,2305,2303,2301,2300,2298,2296,2294,2293,2291,2289,2287
fdb 2286,2284,2282,2280,2279,2277,2275,2273,2272,2270,2268,2266,2265
fdb 2263,2261,2259,2258,2256,2254,2252,2251,2249,2247,2245,2244,2242
fdb 2240,2238,2237,2235,2233,2231,2230,2228,2226,2224,2223,2221,2219
fdb 2217,2216,2214,2212,2210,2209,2207,2205,2203,2202,2200,2198,2196
fdb 2195,2193,2191,2189,2188,2186,2184,2182,2181,2179,2177,2176,2174
fdb 2172,2170,2169,2167,2165,2163,2162,2160,2158,2156,2155,2153,2151
fdb 2149,2148,2146,2144,2143,2141,2139,2137,2136,2134,2132,2130,2129
fdb 2127,2125,2124,2122,2120,2118,2117,2115,2113,2111,2110,2108,2106
fdb 2105,2103,2101,2099,2098,2096,2094,2093,2091,2089,2087,2086,2084
fdb 2082,2080,2079,2077,2075,2074,2072,2070,2068,2067,2065,2063,2062
fdb 2060,2058,2056,2055,2053,2051,2050,2048,2046,2044,2043,2041,2039
fdb 2038,2036,2034,2032,2031,2029,2027,2026,2024,2022,2020,2019,2017
fdb 2015,2014,2012,2010,2009,2007,2005,2003,2002,2000

The bottom line

Look at the assembly code generated by your compiler

Pointers are used to access data structures

Promotion and demotion

Worry about precision and signed when doing math