Fox 11 Firmware

- Firmware
  - Software that is “fixed”
  - Typically found in ROM
  - Common in embedded systems
- Fox 11 Firmware
  - Buffalo Monitor
  - Wytec Phantom Monitor

We use the Wytec Phantom Monitor

1

Buffalo/Wytec Monitor Routines

- Provide support for on-chip/board features
- Primary features
  - COM port routines
    - Sending and receiving data
  - LCD display routines
  - Ability to get date and time (Wytec only)
- See your Fox 11 CD for sample programs

2

Using the LCD Display

- Setup
  - Use **`LCD_INI`** (0x080F)
- Display on top line
  - **`LCD_LINE1`** (0x0812)
    - X points to string, outputs 16 characters
- Display on bottom line
  - **`LCD_LINE2`** (0x0815)
- Others
  - **`SEL_INST, SEL_DATA, WRT_PULSE`**

3

Sample LCD Code

```
.section    .rodata
line1: .asciz  "Hello there, Joe"
; NUL not needed for LCD_LINE1, but might be useful
  in other ways
…
.section    .text
…
lds     #_stack      ; Initialize stack pointer

jsr      LCD_INI     ; Initialize the display

ldx      #line1      ; Display the text
jsr     LCD_LINE1
```

4

Multiply/Divide

- MUL – multiply (unsigned) A x B
  - Result in D
  - 10 clock cycles
- IDIV – integer division, D/X
  - Result in X, remainder in D
  - 41 clock cycles
- FDIV – fractional division, D/X
  - Assumes X > D
  - Result in X as binary fraction, remainder in D
  - 41 clock cycles

5

Binary Fraction?

- What is a binary fraction?
- Binary numbers: 0b11111111 = $2^7(128)+2^6(64)+2^5(32)+2^4(16)+2^3(8)+2^2(4)+2^1(2)+2^0(1)$ = 0xFF = 255
- Binary fraction: 0b11111111 = $2^{-1}(1/2)+2^{-2}(1/4)+2^{-3}(1/8)+2^{-4}(1/16)+2^{-5}(1/32)+2^{-6}(1/64)+2^{-7}(1/128)+2^{-8}(1/256)$ = 0xFF = 0.99609375

6

Fractional Division Example

```
        ldd #0x6000 ; 3/8, numerator
        ldx #0x8000 ; 1/2, denominator
        fdiv
    ; IX = 0xC000 (result: 3/4)
    ; D = 0 (no remainder after 16
    ;         result bits)
```

7

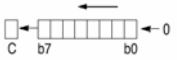
Flag setting

- TST – test, sets/clears Z N flags, V C go to 0
  - tst xxxx – extended mode (no direct or imm.)
  - tst xx,X and tst xx,Y
- TSTA – test A
- TSTB – test B
- SEC / CLC – set / clear carry flag
- SEV / CLV – set / clear overflow flag

8

Shifts

- ASL (= LSL) – arithmetic shift left, ext, indexed, inherent (ASLA, ASLB)

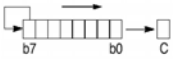


- ASLD (= LSLD) – arithmetic shift left double

9

Shifts

- ASR – arithmetic shift right

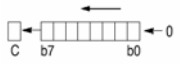


- Why the “recycling” of bit 7?
- Preserves the sign bit (positive or negative)
- ext, indexed, A or B
- No ASRD opcode

10

Shifts

- LSL (same as ASL) – logical shift left, ext, indexed, inherent (LSLA or LSLB)

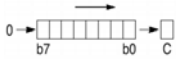


- LSLD (same as ASLD) – logical shift left double

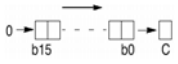
11

Shifts

- LSR – logical shift right, A, B, ext, indexed



- LSRD – logical shift right double

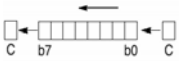


12

Rotates

- ROR – rotate right, ext, indexed, A or B

- ROL – rotate left, ext, indexed, A or B



13