

Embedded RTOS Programming  
Class Project  
m-moore – september 19, 2013

# Features

- GUI-based Monitor/Control of Toy CPU
- All CPU Registers Readable/Writable
- Main-Memory Port-Access
- Program Single-Step
- Program Lock-Step Run
- Load Main Memory From File
- Dump Main Memory To File
- Automatic Test

## RTOS-Related

- Four Threads of Varying Priority
- Mail-Queue Communication Between Threads
- Meta-Watchdog Thread (other threads report-in)
- Selectable `error()` vs `mbed_reset()` on error detection

Meta-watchdog thread.

All other threads check-in via queue.

This thread is also its own watchdog.

```
void watchdogThread(void const *args)    // overall watchdog.
{
    int                dHeartbeat;        // heartbeat counter.
    Watchdog_RTOS watchdog;              // the one and only watchdog.

    dHeartbeat = 0;                       // initialize.

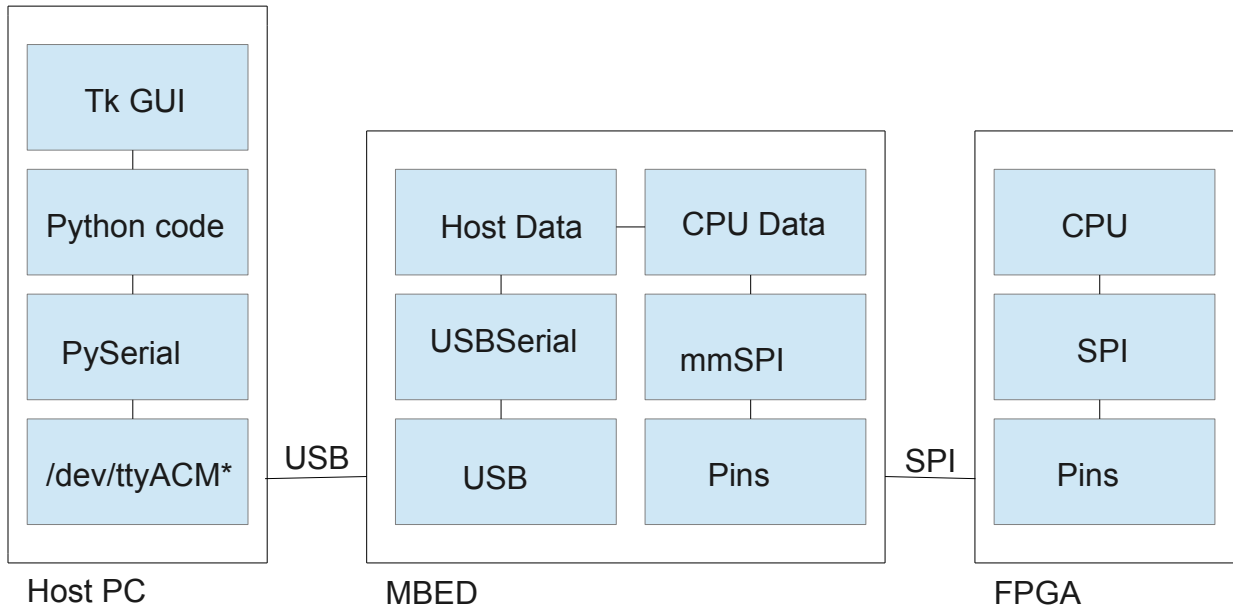
    watchdog.kick(WATCHDOG_S);           // initialize watchdog.

    while (1)                             // thread loop.
    {
                                                // all other threads report-in.
                                                // blocking wait on all of them.
        queueWatchdogThread_0.get(osWaitForever);
        queueWatchdogThread_1.get(osWaitForever);
        queueWatchdogThread_2.get(osWaitForever);

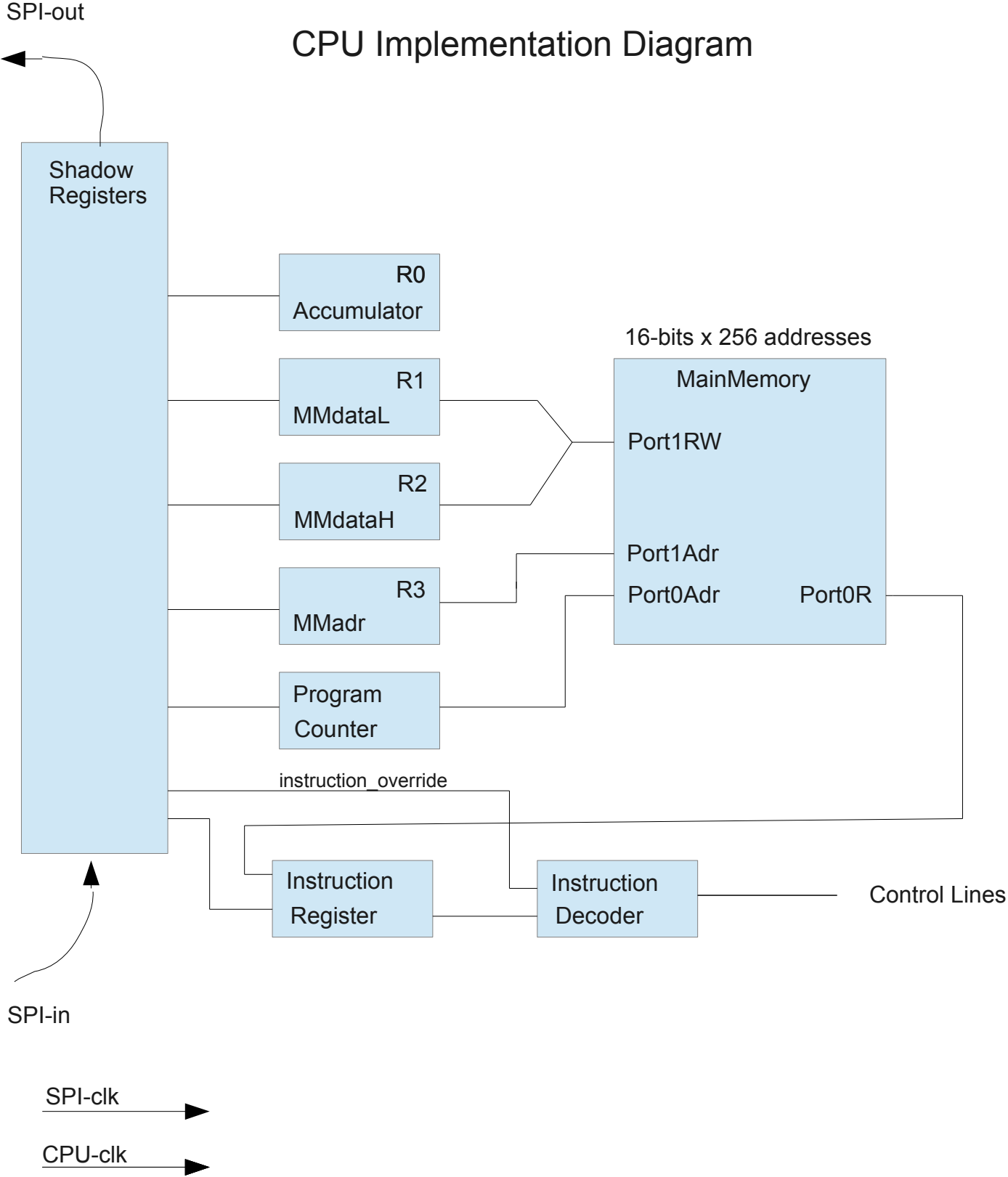
        watchdog.kick();                  // reset watchdog timer.

        dHeartbeat++;                     // thread heartbeat.
        if (!(dHeartbeat % HB_MODULO)) led3 = !led3;
        Thread::wait(THREAD_3_WAIT);      // multitasking.
    }                                     // thread loop.
}
```

# DESIGN DIAGRAM



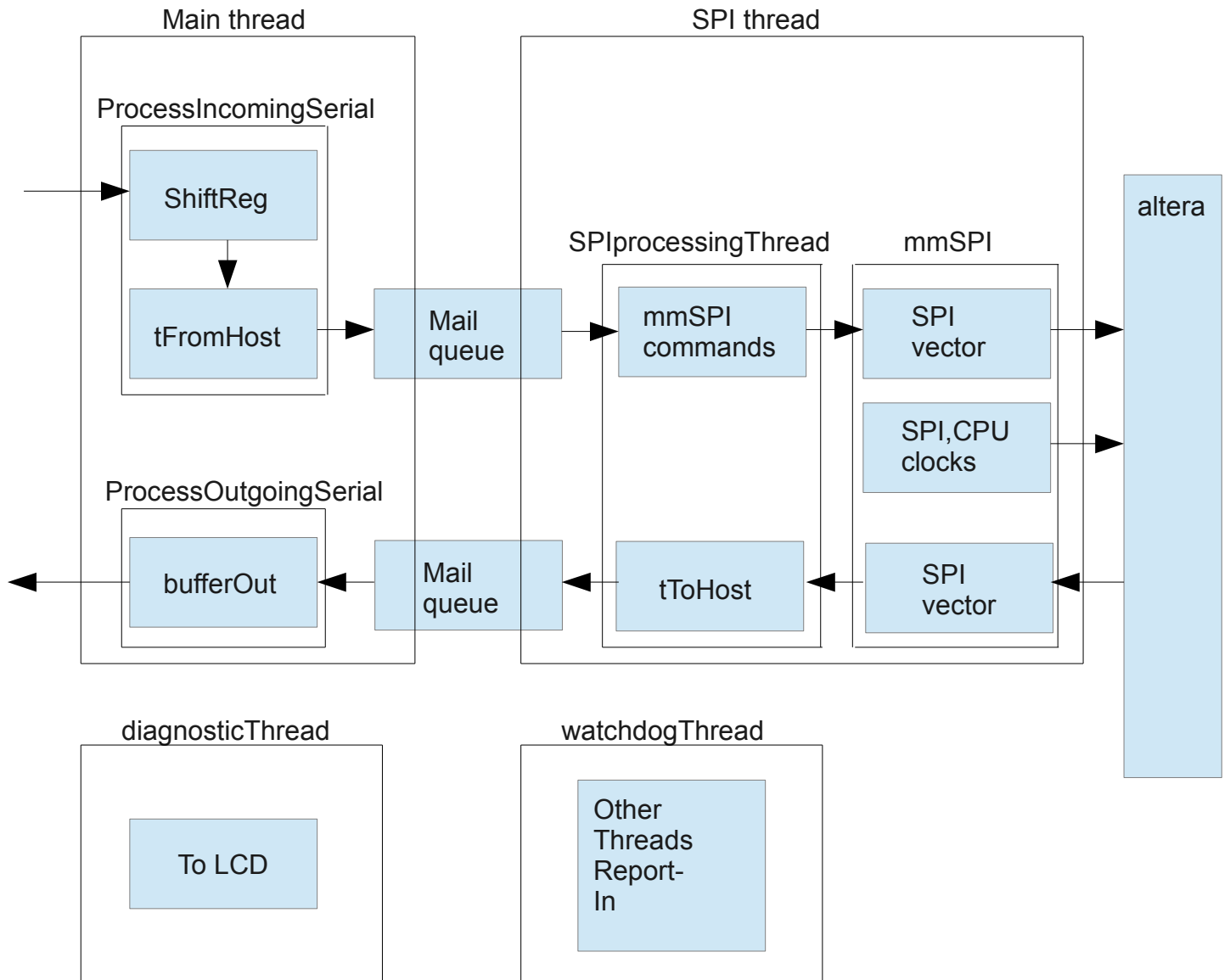
# CPU Implementation Diagram



## Instruction Word Fields

[15:13]	[12:10]	[9]	[8]	[7:0]
SRC	DEST	WE	PCE	Immediate

# Mbed Firmware Implementation Diagram



## Testing

1. debug with LCD & signalTap
2. manual GUI exercise.
3. python speed testing – needed 40mS delay.
4. automated test using 'prog\_add\_list'

Result – no errors detected over a 17-hour run.

419,468,950 SPI clocks.  
6,057,750 CPU clocks.  
17-hour 26-minute test run.  
2050 iterations.  
no errors.

=====

Wed Sep 18 07:43:26 PDT 2013

test iteration 2049

compare dump\_test.txt against dump\_add\_list.txt

=====

Wed Sep 18 07:43:56 PDT 2013

test iteration 2050

compare dump\_test.txt against dump\_add\_list.txt