Recap

Logical operations (set, clear, mask, toggle)
Shift operations
Arithmetic operations (CCR bits)

Overview

Debugging
TExaS
Real 9S12DG128

	9S12DG128	9S12DP512
I/O	$0000 to $03FF	$0000 to $03FF
Ports	A,B,E,H,K,J,P,AD0,AD1, M, S, T	A,B,E,H,K,J,P,AD0,AD1, M, S, T
RAM	$2000 to $3FFF, 8 KiB	$0800 to $3FFF, 14 KiB
ROM	$4000 to $FFFF, 48 KiB	$4000 to $FFFF, 48 KiB

Good data sheets to have available
32 page CPU12 quick reference
http://users.ece.utexas.edu/~valvano/Datasheets/CPU12rg.pdf
458 page CPU12 programming reference
http://users.ece.utexas.edu/~valvano/Datasheets/S12CPUV2.pdf
datasheets from Freescale
http://users.ece.utexas.edu/~valvano/Datasheets/MC9S12DP512.pdf
In EE319K, the only difference between 9S12DP512 and 9S12DG128 is the size of the RAM

Debugging terms that essential mean the same thing

- Testing
- Debugging
- Diagnostics
- Verification

Debugging Approach

- Control, defining input values, repeated measurements
- Observability,
  - Black box, see just output values
  - White box, see internal parameters and output values

Debugging Actions

- Functional debugging, input/output values
- Performance debugging, input/output values with time
- Tracing, measure sequence of operations
- Profiling,
  - measure percentage for tasks,
  - time relationship between tasks
- Performance measurement, how fast it executes
- Optimization, make tradeoffs for overall good
  - improve speed,
  - improve accuracy,
  - reduce memory,
  - reduce power,
  - reduce size,
  - reduce cost

Debugging Items

- Instrumentation, code with add to system
- Visualization, how we see debugging information

Goal of debugging

maintain and improve software
remedy faults or to correct errors in a program
role of a debugger is to support this endeavor

The debugging process

testing,	Finding bugs
stabilizing,	Fix inputs, so it can be run over and over
localizing, and	Solving a simplier problem
correcting errors.	Fixing bugs and repeat

"rough and ready" manual methods

desk-checking	Think about it with paper and pencil
dumps,	Show inputs, intermediates, and outputs
print statements	

Intrusiveness

degree of perturbation caused by the debugging itself
how much the debugging slows down execution

Nonintrusive

characteristic or quality of a debugger
allows system to operate as if debugger did not exist
e.g., logic analyzer, ICE, BDM

Minimally intrusive

negligible effect on the system being debugged
e.g., dumps (ScanPoint) and monitors

Highly instrusive

e.g., print statements, breakpoints and single-stepping

TExaS debugging features to demonstrate

uc – ViewBox, BreakPoints
stk – StackField, MemoryBox
Modes
- **FollowPC**
- **CycleView**
- **InstructionView**
- **LogRecord**

Single Step, Few, StepOver, StepOut, Run
Breakpoint versus ScanPoint

Run time errors

Read from unprogrammed ROM
Write to ROM
Read from unitialized RAM
Read/write unimplemented I/O
A bug in your program (EE319K)
or I/O port not supported by TExaS (SPI in EE345L)

A debugging instrument

software code that is added to the program
the purpose of debugging
e.g., print statement
using editor/assembler/loader, one adds instrument

- Place all instruments in a unique column
- Define instruments with specific pattern in their names
- Use instruments that test a run time global flag
  leaves a permanent copy of the debugging code
  causing it to suffer a runtime overhead
  simplifies "on-site" customer support.
- Use conditional compilation (or conditional assembly)
  TExaS does not support conditional assembly
  Easy to remove all instruments

Download and unzip http://users.ece.utexas.edu/~valvano/Starterfiles

```
      Simple_DP512asm.zip
      Square_DP512asm.zip
      NotGate_DP512asm.zip
```

Heart Beat LED monitor (PP7)

Question: Is my program running?
Method: Add instruments, assemble, download, run
Initialization

```
   bset DDRP,#$80             DDRP |= 0x80;
```

Debugging Instrument (toggle PP7)

```
   ldaa PTP                         PTP ^= 0x80;
   eora #$80
   staa PTP
```

Debugging Instrument (set PP7)

```
   bset PTP,#$80                    PTP |= 0x80;
                                                PTP_PTP7 = 1;
```

Debugging Instrument (clear PP7)

```
   bclr PTP,#$80                    PTP &= ~0x80;
                                                PTP_PTP7 = 0;
```

Global dump

Question: Is my program running?
Method: Add instruments to NotGate2, assemble, download, run
Initialization

```
;add this variables to global RAM
Count rmb 1            unsigned char Count;
```

Debugging Instrument (called once)

```
    clr Count                 Count = 0;
```

Debugging Instrument (how many times)

```
    inc Count       Count++;
```

Global dump

Question: Is my program getting input or producing output?
Method: Add instruments, assemble, download, run
Initialization

```
;add this variable to global RAM
Data  rmb 2
```

Debugging Instrument (save a copy)

```
    stx Data
```

To use TExaS with the real 9S12 board you will need version 1.37.
1) It is good design practice to develop and test the software first using the simulator. You will need at least a UC and RTF file. Within TExaS click the slider Mode switch on UC window to the "Simulator" position.
2) To connect the board do these steps
  - -Cut out a piece of paper and place it between the 9S12 and the breadboard as you plug the 9S12 board into the breadboard.
  - -Make all hardware connections and disconnections while power is off
  - -Plug the serial cable between the 9S12 and the PC (if you know the COM port number, it will be good)
  - -Attach the power adapter plug into the 9S12 and apply power
  - -Place the 9S12 into LOAD mode and hit the reset button

3) Execute Mode->RunMode… and set the COM port connection to match your COM port. You can enter a "0" and TExaS will search COM1 COM2 COM3… looking for the 9S12 board. Within this dialog set the "Time Out Delay" to 100 ms. Other setting in this dialog can be adjusted as you wish. Click OK.
4) Within TExaS click the slider Mode switch on UC window to the "Real 9S12" position.
5) Click on the RTF source file and assemble the program. View the activity in the TheLog.rtf window to verify the TExaS can communicate with the board.

Run Simple in embedded system mode (8 MHz)
Quit debugger
Remove power
Remove RS232 cable
Switch to RUN mode
Apply power
Is your PP7 light flashing three times slower?

The bottom line
Monitor is a real-time visualization (heartbeat)
Dump records data for later analysis
Simulation (test), prototype (test), final system (test)