

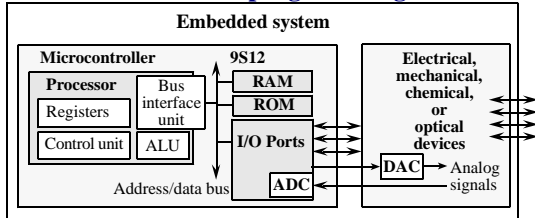
Recap

- Embedded system
- Microcontroller
- Data Flow graph

Overview

- 9S12 programming
- TEaS simulator
- Top down design

Introduction to 9S12 programming



Register A, 8-bit number from 0 to 255

Z-bit, 0 means result not zero, 1 means result is zero

\$0240 is Port T, input/output port

\$0242 is the direction register for PortT (0=in, 1=out)

\$0800 to \$3FFF is RAM, volatile, variables (DP512)

\$4000 to \$FFFF is EEPROM, nonvolatile, program (DP512)

```

;template
    org    $4000
main

    stop
    org    $FFFE
    fdb    main
    
```

Short list of instructions to get started

```

ldaa #10    ;make A=10
ldaa $0800 ;A=contents of memory
staa $0800 ;store contents of A into mem
deca       ;A=A-1
lsra       ;A=A/2 (shift right)
lsla       ;A=A*2 (shift left)
adda #10   ;A=A+10
anda #$02  ;A=A&2 (logic and)
eora #$08  ;A=A^8 (exclusive or)
oraa #$03  ;A=A|3 (logic or)
bra loop   ;always jump to loop
bne loop   ;jump to loop if not zero
beq loop   ;jump to loop if zero
stop
    
```

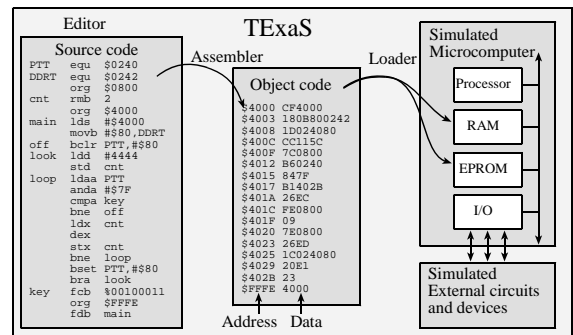
Appendix 1. Embedded system development using TEaS

A1.1. Introduction to TEaS

Figure 2.1. Assembly language development process in TEaS.

TEaS supports six phases of software development:

- Defining the microcomputer type and memory configuration,
- Writing the program source code using an editor,
- Assembling source code and loading object code into memory,
- Interfacing external components,
- Debugging the program by running it on the interactive simulator
- Debugging the program by running it on an actual board



Observation: Hiding windows will improve the simulation speed.

Checkpoint A1.1: What is an assembly source code?

Program file (*.rtf) source code.

TheLog.rtf logs information, interactive debugger
 TheList.rtf the assembly listing
 TheCRT.rtf the input/output data of a CRT terminal

Microcomputer file (*.uc) internal microcomputer

I/O Device file (*.io) external I/O devices

switches, LEDs, LCDs, keyboard, the CRT, motors, IR and sensors.

Stack file (*.stk) holds temporary information

Scope file (*.scp) used for debugging

Plot file (*.plt) display graphical information

Checkpoint A1.4: What is object code?

A1.2. Major components of TExaS

Editor

Assembler

Source code to object code

Simulation Mode

Instruction set simulator

bus cycle activity and the extensive error checking

I/O port simulator

External device simulator

Real mode

Debugger interface to real 9S12 board

Help system

How TExaS works

9S12 reference material

Instructions

I/O ports

Lots of examples

Unique aspect of this simulator is the error checking.

- Execution of an illegal instruction,
- Read/write to an undefined address,
- Stack underflow (causing a read/write from unimplemented memory),
- Write to ROM, EPROM, EEPROM,
- Read from unprogrammed ROM, EPROM, EEPROM,
- Read from RAM that has not yet been written to,
- Read from an unimplemented I/O port.

Figure 1.5. Product development cycle.

Top down embedded system design

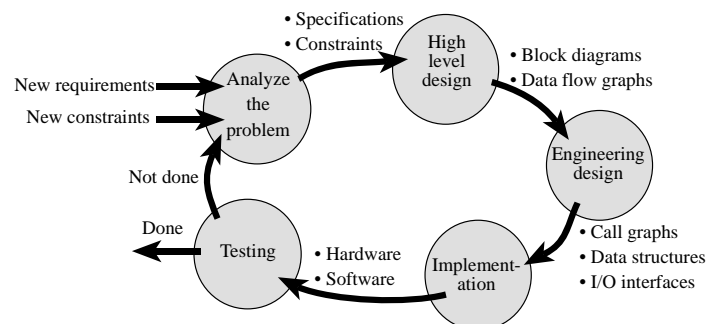
first draw a data flow graph

choose inputs, outputs, data structures

define major hardware/software modular blocks

estimate calculations/sec, memory size

choose a processor family



next draw a call graph showing the control linkage
 design at a very high level using pseudo code
design at a lower level using a high level language like C
 convert the software by hand into assembly
 simulate prototype
 test and redesign algorithms
 choose a specific microcontroller
 build and test actual prototype

Design Problem – NOT gate

Steps

Market survey, customer focus group, cost/benefit analysis

Inputs, outputs (specifications, constraints)

Data flow graph

Design for test (add features to facilitate testing)

Flowchart – pseudo code

Edit source code

Build input/output mechanical/electrical devices in simulator

Interface to 9S12 in simulator

Debug system in simulator

Design for test (add features to facilitate testing)

Build real input/output devices

Interface real devices to 9S12

Debug real 9S12 prototype

Redesign for production

 Design for test (add features to facilitate testing)

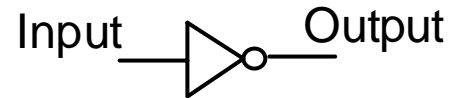
 Manufacturability (can it be built?),

 Maintainability (can it be repaired?),

 Reliability (how long will it work?)

Power, packaging, safety, component availability

Test production units



Online homework submission

You can work together in groups of any size, but everyone enters a separate online solution. Other than breaking into the system and changing grades, you may get as much help from anyone in any way you please to complete the online homework.

STEP 1: Log into the Quest Homework Service at the URL <https://quest.cns.utexas.edu/student/> Click Get Started, log into with UT EID, select EE319K

STEP 2:

Download: Students' Instructions

Download: First Homework

STEP 3: Work one homework question. Log in again and submit its answer before next class period.

STEP 4: Continue submitting answers until due time.

STEP 5: Download the solutions after due time.

The bottom line

You've learned assembly programming in EE306

Get the reference materials on 9S12 instructions

Bosses love top-down design

TEaS simulates hardware and software

TEaS will be also used to download onto real board

Metrowerks will allow projects and C code