Appendix: A

List of publication based on Research work:


3. B. C. Goradiya, Dr. H.N.Pandya, V.V.Kamdar “Embedding RTOS on ARM 7 Architecture” in National Conference on Recent Trends in Engineering & Technology at BVM Engineering College, V.V.Nagar, Gujarat, India. 13-14 May 2011. ISBN: 978-81-921358-3-0

4. B. C. Goradiya, Dr. H.N. Pandya “Enhancing ARM core capability using PSoC5” in International Journal of Engineering Research & Technology (IJERT) Volume 1 Issue 5, July 2012. ISSN: 2278-0181


Appendix: B

References:

2. The official website of ARM limited www.arm.com
3. ARM System-On-Chip architecture by Steve Furber – Pearson publication.
4. High-Level Optimization for Low Power Consumption on Microprocessor-Based Systems by David A. Ortiz, Nayda G. Santiago
5. Power analysis and minimization techniques for embedded DSP software by M.T. Chien Lee, V. Tiwari, S. Malik, M. Fijita
6. Power Reduction Techniques for Microprocessor Systems by VASANTH VENKATACHALAM AND MICHAEL FRANZ

14. www.simplescalar.com


17. LPC2138 controller specification, www.NXP.com

18. RTOS guidelines, www.freertos.com


28. Cypress PSoC Creator 1 Manual from Cypress.com

29. Operational Manual of the Cypress CY8CKIT001 development kit

30. Application note from cypress.com on AN58304 -PSoC®3 and PSoC5 - Pin Selection for Analog Designs
31. Application note from cypress.com on AN77900 -PSoC®3 and PSoC5 Low-power Modes and Power Reduction Techniques


40. https://github.com/raspberrypi

41. https://github.com/raspberrypi/linux

42. http://www.arduino.cc

43. http://www.keil.com

Appendix: C

Steps for writing Raspbian OS on SD card other than windows OS:

Ubuntu:

2. Install the usb-imagewriter package
   1. If your release does not include this, download it from Oliver’s PPA
   2. If imagewriter fails to launch, you may need to install python glade2 support. Install the python-glade2 package or Run
      sudo apt-get install python-glade2
   3. If your release does not include it and you are running 9.04 Jaunty Jackalope then run this command from the console:
      sudo apt-get install usb-imagewriter
3. Open Applications -> Accessories -> Image Writer
   1. KDE users will find this in Applications -> Utilities -> Image Writer
   2. from the command line, from the console:
      sudo imagewriter
   3. on some usb-imagewriter versions (console command: imagewriter) the application fails to write if the image path contains blank spaces, exiting with "IndexError: list index out of range".
4. Insert your flash media
5. Select the downloaded file and flash device, and click "Write to Device"
6. Remove your device when the operation is complete

Graphical Interface for Linux
Command Line Interface:

Be very careful about which /dev device you write to. If your machine is booted up off of disk /dev/sda, and your usb stick is on /dev/sdc, and you accidentally write to /dev/sda instead of /dev/sdc, your filesystem will be irreparably damaged and you will lose all of your files.

2. Open a terminal and insert your flash media
3. Look at the output of 'sudo dmesg | tail -20' to determine the device node assigned to your flash media (ignore the device number; e.g. /dev/sdc, not sdc1)
4. Run
   ```
   sudo umount /dev/devicenode
   ```
5. Run
   ```
   sudo dd if=/path/to/downloaded.img of=/dev/devicenode bs=1M
   ```
6. Remove your flash media when the command completes (you may need to wait a few extra seconds for it to finish)

Mac OS X:

Command Line Interface:

2. Open a Terminal (in /Applications/Utilities/)
3. Run ‘diskutil list’ to get the current list of devices
4. Insert your flash media
5. Run ‘diskutil list’ again and determine the device node assigned to your flash media (e.g. /dev/disk2)
6. Run
   ```
   diskutil unmountDisk /dev/diskN (replace N with the disk number from the last command; in the previous example, N would be 2)
   ```
7. Execute
   ```
   sudo dd if=/path/to/downloaded.img of=/dev/rdiskN bs=1M (replace /path/to/downloaded.img with the path where the image file is located; for example, ./ubuntu.img, /dev/rdiskN is faster than /dev/diskN). If you see the error dd: Invalid number `1m', you are using GNU dd. Use the same command but replace bs=1m with bs=1M.
   ```
8. Run `diskutil eject /dev/diskN` and remove your flash media when the command completes.
### Appendix: D

**Code density comparison for 16-bit multiplication:**

<table>
<thead>
<tr>
<th>8-bit example: based on 8051</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOV A, XL ; 2 bytes</td>
</tr>
<tr>
<td>MOV B, YL ; 3 bytes</td>
</tr>
<tr>
<td>MUL AB ; 1 byte</td>
</tr>
<tr>
<td>MOV R0, A ; 1 byte</td>
</tr>
<tr>
<td>MOV R1, B ; 3 bytes</td>
</tr>
<tr>
<td>MOV A, XL ; 2 bytes</td>
</tr>
<tr>
<td>MOV B, YH ; 3 bytes</td>
</tr>
<tr>
<td>MUL AB ; 1 byte</td>
</tr>
<tr>
<td>ADD A, R1 ; 1 byte</td>
</tr>
<tr>
<td>MOV R1, A ; 1 byte</td>
</tr>
<tr>
<td>MOV A, B ; 2 bytes</td>
</tr>
<tr>
<td>ADDC A, #0 ; 2 bytes</td>
</tr>
<tr>
<td>MOV R2, A ; 1 byte</td>
</tr>
<tr>
<td>MOV A, XH ; 2 bytes</td>
</tr>
<tr>
<td>MOV B, YL ; 3 bytes</td>
</tr>
<tr>
<td>MUL AB ; 1 byte</td>
</tr>
<tr>
<td>ADD A, R1 ; 1 byte</td>
</tr>
<tr>
<td>MOV R1, A ; 1 byte</td>
</tr>
<tr>
<td>MOV A, B ; 2 bytes</td>
</tr>
<tr>
<td>ADDC A, R2 ; 1 byte</td>
</tr>
<tr>
<td>MOV R2, A ; 1 byte</td>
</tr>
<tr>
<td>MOV A, XH ; 2 bytes</td>
</tr>
<tr>
<td>MOV B, YH ; 3 bytes</td>
</tr>
<tr>
<td>MUL AB ; 1 byte</td>
</tr>
<tr>
<td>ADD A, R2 ; 1 byte</td>
</tr>
<tr>
<td>MOV R2, A ; 1 byte</td>
</tr>
<tr>
<td>MOV A, B ; 2 bytes</td>
</tr>
<tr>
<td>ADDC A, #0 ; 2 bytes</td>
</tr>
<tr>
<td>MOV R3, A ; 1 byte</td>
</tr>
</tbody>
</table>

**Code size:** 48 bytes  
**Time to execute:** 48 clock cycle

<table>
<thead>
<tr>
<th>16-bit example:</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOV R4,&amp;0130h</td>
</tr>
<tr>
<td>MOV R5,&amp;0138h</td>
</tr>
<tr>
<td>MOV SumLo,R6</td>
</tr>
<tr>
<td>MOV SumHi,R7</td>
</tr>
</tbody>
</table>

(Operands are moved to and from a memory mapped hardware multiply unit)  
**Code size:** 8 bytes  
**Time to execute:** 8 clock cycle

<table>
<thead>
<tr>
<th>ARM Cortex-M:</th>
</tr>
</thead>
<tbody>
<tr>
<td>MULS r0,r1,r0</td>
</tr>
</tbody>
</table>

ro and r1 assume to carry 16-bit data they are able to perform multiplication of 32 bits  
**Code size:** 2 bytes  
**Time to execute:** 1 clock cycle