CEG2400 - Microcomputer Systems

Lecture 11: Thumb mode and Case Studies

Philip Leong

Introduction

- A number of quizzes for you to test your understanding of the material so far
- Thumb mode
- Embedded system case study
  - How we use the microprocessor stuff in consumer electronics products

I/O Quiz

- A serial device transmits characters at 9600 characters per second, and it takes 5 uS to process an interrupt. What is the amount of CPU overhead per second for this device if it is
  - Interrupt driven
  - DMA driven. Assume that the DMA controller transfers 1000 bytes at a time, and that memory words are 32-bit with an access time of 10 ns. Assume that transfers are buffered and transmitted a word at a time.

I/O

- Interrupt:
  - Every second transmit 9600 bytes
  - 9600 bytes x 5 uS/byte = 48ms
- DMA:
  - Every second transmit 9600 bytes
  - For 1000 bytes: 1000/4=250 transfers at 10ns each = 2.5uS. Also have one interrupt 5 uS.
  - so for 9600 bytes: 9.6 x (2.5+5)=72 uS

Pipelining Quiz

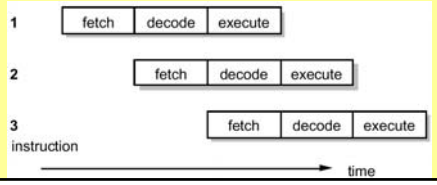
- Suppose that we have an ARM7 device with 3 stage pipeline. Draw a diagram illustrating execution of the following program. How many cycles does it take to complete?

```
mov r0,#0
mov r1,#1
mov r2,#2
```

- Suppose we have a sequence of n mov statements similar to the above, how many cycles does it take to complete?
- How many cycles would it take if no pipelining was used?
- Suppose that without pipelining the maximum clock rate is 50MHz, what would you expect the pipelined clock rate to be? Explain your answer.

Pipelining

- Overall, takes 5 cycles
- For N instructions, we need N+2 cycles
- For no pipelining, we have 1 instruction per cycle so it takes 3 cycles
- Clock rate should be able (50*3)MHz as each pipeline stage does about 1/3 as much work. In practice, it may be difficult to divide the work into 3 exactly equal tasks



ARM Quiz

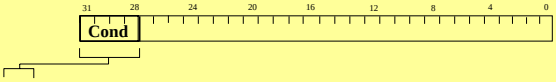
- For each of the ARM arithmetic instructions below, write the operation that the ALU must perform
  - ADD,ADC,SUB,SBC,RSB,RSC

ARM

- Operations are:
  - ADD operand1 + operand2
  - ADC operand1 + operand2 + carry
  - SUB operand1 - operand2
  - SBC operand1 - operand2 + carry -1
  - RSB operand2 - operand1
  - RSC operand2 - operand1 + carry - 1

ARM Quiz

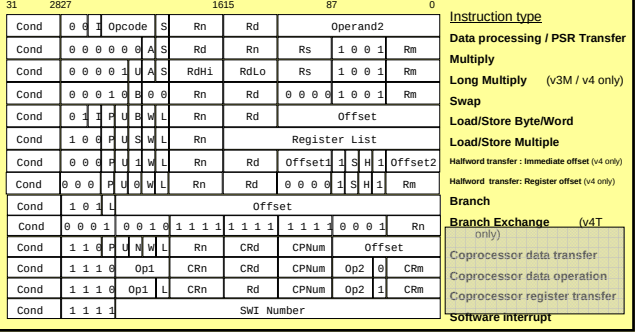
- For a BLS instruction, what is the binary code for the Cond field



0000 = EQ - Z set (equal)
0001 = NE - Z clear (not equal)
0010 = HS / CS - C set (unsigned higher or same)
0011 = LO / CC - C clear (unsigned lower)
0100 = MI -N set (negative)
0101 = PL - N clear (positive or zero)
0110 = VS - V set (overflow)
0111 = VC - V clear (no overflow)
1000 = HI - C set and Z clear (unsigned higher)
1001 = LS - C clear or Z (set unsigned lower or same)
1010 = GE - N set and V set, or N clear and V clear (>or =)
1011 = LT - N set and V clear, or N clear and V set (>)
1100 = GT - Z clear, and either N set and V set, or N clear and V set (>)
1101 = LE - Z set, or N set and V clear,or N clear and V set (<, or =)
1110 = AL - always
1111 = NV - reserved.

ARM Quiz

- What is the machine code for the "SWI 1" instruction? What is the max address for a branch to a label? How can we branch further?



Instruction type
Data processing / PSR Transfer
Multiply
Long Multiply (v3M / v4 only)
Swap
Load/Store Byte/Word
Load/Store Multiple
Halfword transfer : Immediate offset (v4 only)
Halfword transfer: Register offset (v4 only)
Branch
Branch Exchange (v4T only)
Coprocessor data transfer
Coprocessor data operation
Coprocessor register transfer
Software interrupt

ARM

- BLS – 1001
- 0xef000001
- From the table, 24 bits are available so the largest jump is 0x7fffff (otherwise it is a negative number)
  - i.e. the max address to which we can branch is pc+0x7fffff
  - Another way is to use "BX reg" or "mov pc,reg" which can jump to an arbitrary location

Thumb mode

Introduction

- ARM code is 32-bit, 3-address code
  - High speed
  - Orthogonal instruction set
- Often we wish to improve code density (i.e. make code fit into smaller memory)
- Thumb mode addresses this issue
  - Extension to the ARM architecture
  - Achieves approx 30% improvement in code density

Thumb/ARM differences

- Thumb
  - 16-bit instructions
  - No conditional instructions
  - 2-address format
  - Less regular
  - 8 registers (a few instructions allow r8-r15 to be specified)
- ARM
  - 32-bit instructions
  - ARM instructions are conditional
  - 3-address format
  - Regular instruction set
  - 16 registers

Fewer Registers

- How does this aid code density?

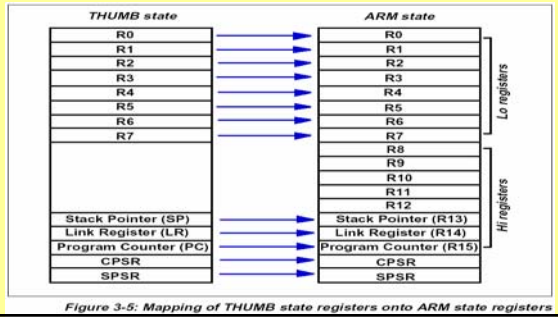
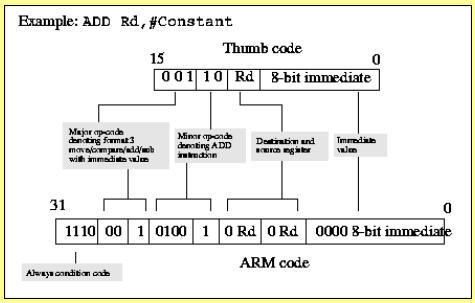


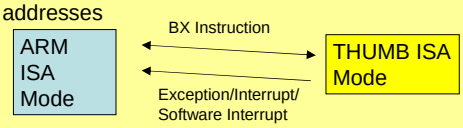
Figure 3-5: Mapping of THUMB state registers onto ARM state registers

Thumb is subset of ARM



Thumb to ARM

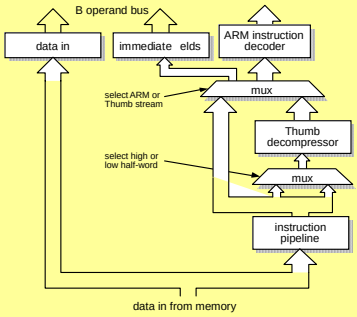
- 16-bit code
  - Switch to ARM ISA mode automatically on exception
  - Switch by BX or BLX (Branch and Exchange or Branch and Link with Exchange) instruction
    - (T bit in CPSR set to bottom bit of the address)
    - ARM/Thumb instructions must always branch to even addresses



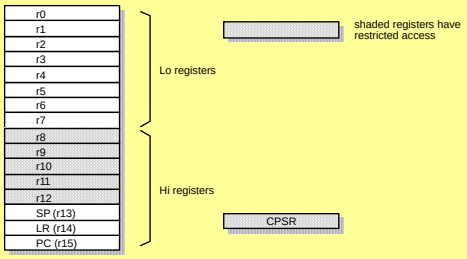
Instruction Set

ADC	ADD	AND	ASR
B	BCC	BIC	BL
BX	CMN	CMP	EOR
LDMIA	LDR	LDRB	LDRH
HSL	LDRSB	LDRSH	HSR
MOV	MUL	MVN	NEG
ORR	POP	PUSH	ROR
SBC	STMIA	STR	STRB
STRH	SWI	SUB	TST

Thumb Implementation

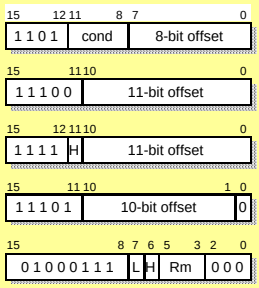


Accessible Registers

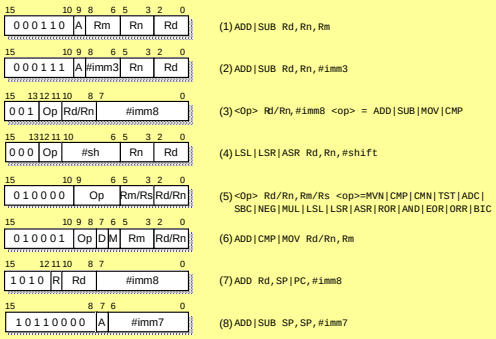


Instructions

Branch Type

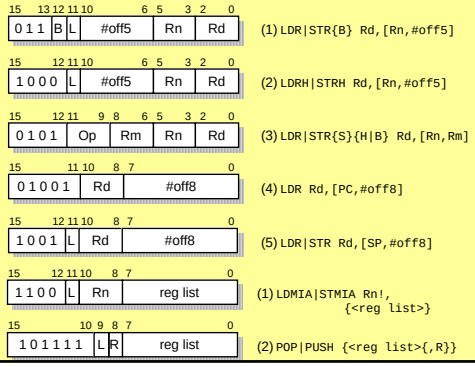


Data Processing



Note that often a 2-address format used.

Register Transfer



- ARM

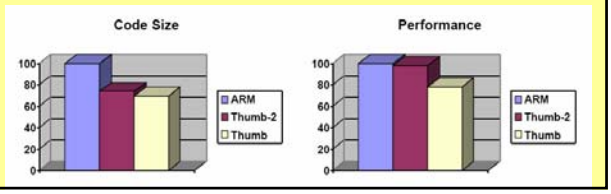
```
LDREQ r0,[r1]
LDRNE r0,[r2]
ADDEQ r0, r3, r0
ADDNE r0, r4, r0
```

- Thumb

```
   BNE L1
   LDR r0, [r1]
   ADD r0, r3, r0
   B L2
L1 LDR r0, [r2]
   ADD r0, r4, r0
L2
```


Code density and Power

- Thumb instructions take up less memory but we need slightly more of them to do the same thing
  - ARM mode: Gives best performance and hence may be able to clock an application at lower rate to save power
  - Thumb mode: Better density may result in less memory traffic and hence reduce power
  - Not a straightforward relationship (depends on application)
- A mix of the two is common (ARM for performance sensitive code, Thumb for rest)



Thumb Quiz

- What is the advantage of the Thumb instruction set?
  - What are the disadvantages?
- What is done in the Thumb instruction set to improve code density?
- Give examples of 2-address and 3-address instructions
- Suppose we have 2 processors, A and B. Both operate at the same clock rate and with the same continuous power dissipation but A can execute a given program in 20% less time than B. Assuming that power is only consumed while the program is running, how much longer battery life would A have compared with B.

Thumb

- Advantages: improved code density
  - Disadvantages: lower execution speed, slightly larger die size
- Fewer instructions, two address instructions, smaller register file set, no conditionals
- lsl r0,r1,r2 vs lsl r0,r1 (add is 3-address in Thumb)
- Battery life equates to energy which is power * time. Since time is reduced by 20%, so is energy.

Case Study: Apple 3G ipod

Source: http://members.chello.nl/~m.heijligers/ipod/Engineering/engineering.html
http://www.designchain.com/coverstory.asp?issue=summer02

Design Criteria

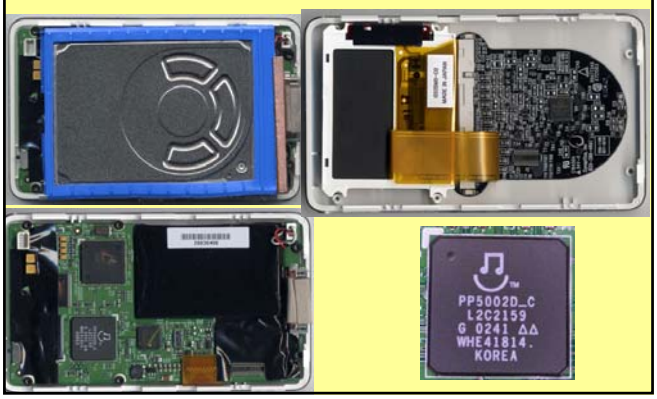
- Highest quality sound
- Off-the-shelf components
- Cost
- Time to market

- Many consumer electronics products have similar goals e.g. mobile phones, cameras, etc

Design

- A lot of effort placed in outside appearance, user interface and small volume
- Sony battery, 1.8 in Toshiba drive and PCB are layered

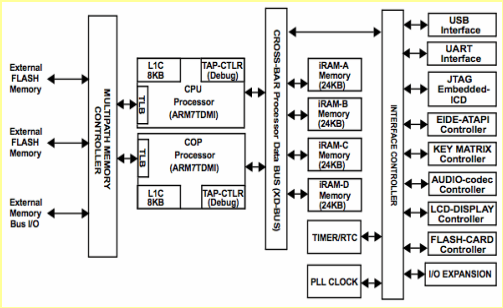
Internals 3G 30GB ipod



Risk

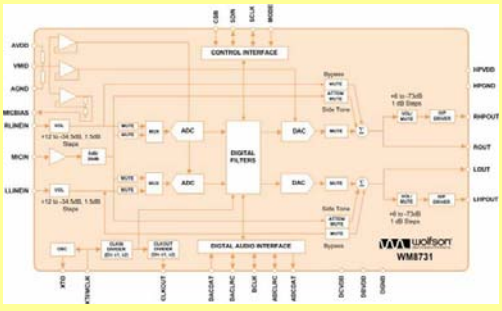
- Time-to-market and risk management
  - A custom integrated circuit would have given them improved cost per unit, volume and battery life
  - Increased risk of design flaw which would affect time to market
  - They decided best to choose the best third-party components (higher quality, shorter time to market, lower risk)
- Portalplayer has development environment for customized, high quality audio player
  - Also have reference design (can buy PCBs and working software for their products)
- Apple
  - Put everything together, developed user interface, optimized for performance, price etc. This is the most important step!

Portalplayer PP5002



Audio Output

- Wolfson codec with integrated headphone driver



Other Chips

- Texas Instruments Firewire Driver.
- Samsung 32MB BGA SDRAM chip, which serves as a buffer for songs, not having to acces the HD constantly.
- Philips Semiconductors Power Management and DC-to-DC convertor ICs.
- Cypress Semi USB controller.
- Sharp provides a1MB Flash for firmware.
- TI bridge interface to control the HD.

Other Components

- Toshiba provides the 1.8" hard disk (for the iPod mini, this is Hitachi), approx 50% of total cost
- The touchpad wheel and controller is made by Synaptics. More information can be found at mp3.com.
- Quadros Systems provides the real-time OS.

Key points

- Most embedded systems (computer-based system that performs a dedicated function) companies (e.g. Apple) buys the components and develops products based on it
  - Have a range of products: ipod shuffle, nano, hard disk version
  - Volume production in China