EEE3096S: Embedded Systems II

LECTURES 7-8:

ARM ARCHITECTURE

Presented by:

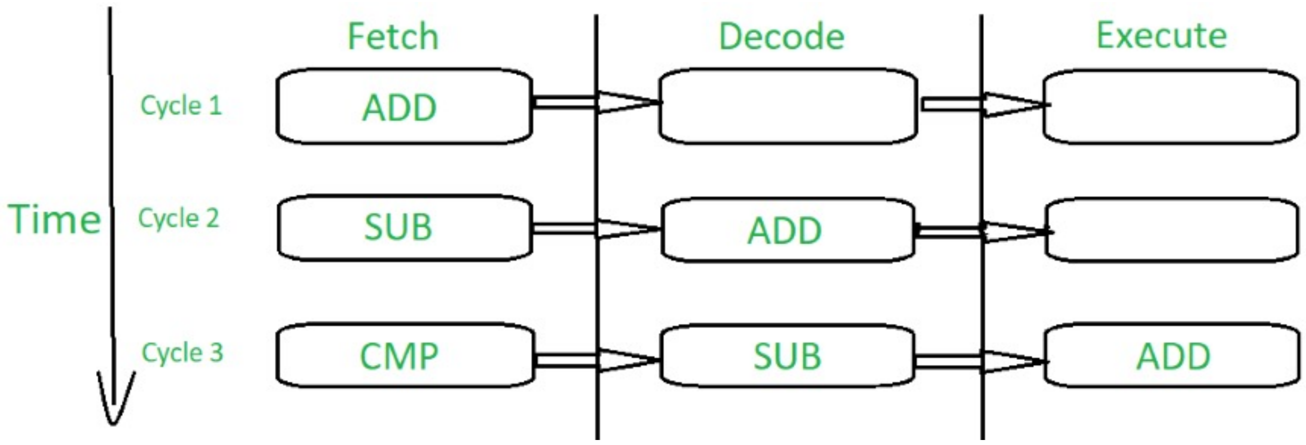
Dr Yaaseen Martin

Electrical Engineering
University of Cape Town

BACK TO ARM…

THE ARM CORTEX GENERATIONS

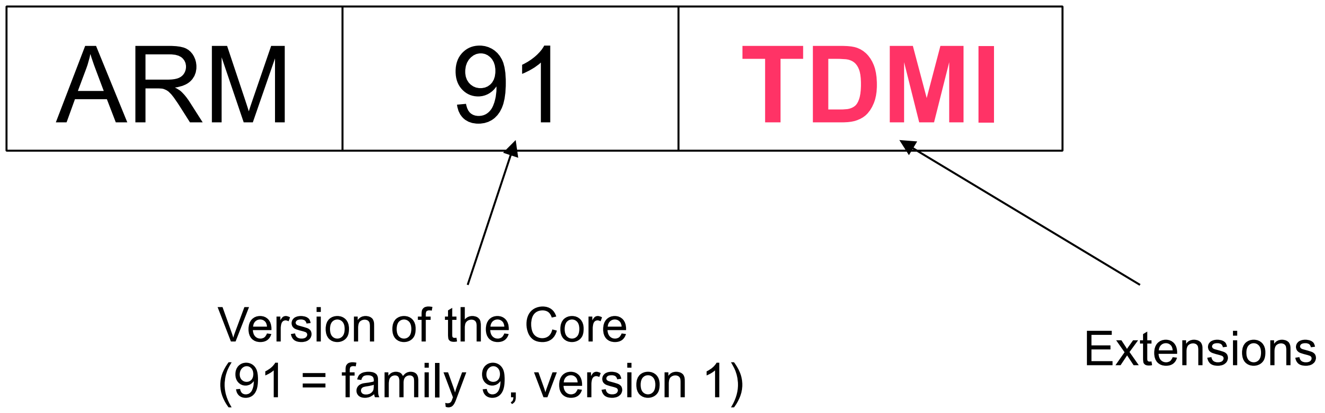
- ARM Cortex range introduced in 2005
- Key features of the Cortex-M0:
  - ARMv6-M architecture
  - High performance relative to power consumption
  - Some [Thumb]() instruction set support
  - 3-stage pipeline



WHY THUMB INSTRUCTIONS?

- Thumb halves the size of the instructions; uses **16-bit instructions** on 32-bit system
- System can load two Thumb instructions in one data word and process them in the correct order
- **But** at the expense of complexity/functionality of the instruction set
- For many applications, code density can be nearly doubled, increasing performance

ARM CORE NAMING CONVENTION



Some extensions:

E	DSP instruction extensions
J	Jazelle, Java bytecode extensions
T	Thumb – 16-bit operations supported
D	Debugger support

DATA AND INSTRUCTIONS

- ARM 32-bit architecture termed “**AArch32**”
- Support operations on bytes (8-bit), half-words (16-bit) and words (32-bit).
- All cores support 32-bit ARM instructions
- Most cores (ARM7 and later) support 16-bit **Thumb instructions**
- Some (Jazelle) cores support 8-bit **Java bytecode**

THE 7 MODES OF OPERATION

1. **User Mode**
   - Restricted mode – limitations on register access and memory addressing.
   - Applications generally run in this mode.
2. **Interrupt ReQuest (IRQ)**
   - This mode is used when a low priority interrupt is signalled.

THE 7 MODES OF OPERATION

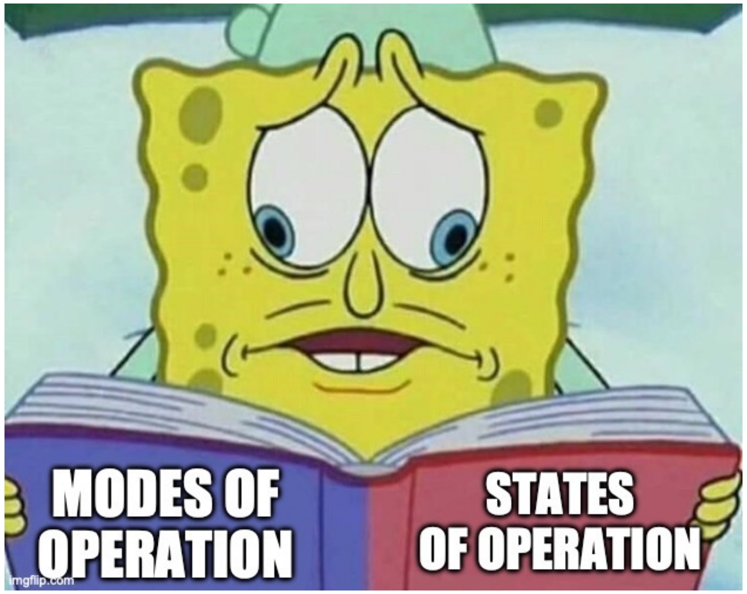
3. **Fast Interrupt reQuest (FIQ)**
   - Used when a high priority interrupt is signalled; separate set of registers
   - Handled differently to a regular IRQ
4. **Supervisor**
   - Entered on system reset and on software interrupt instruction
5. **Abort**
   - Memory exceptions handled in this mode

THE 7 MODES OF OPERATION

6. **Undef**
   - Handles undefined instructions
7. **System Mode**
   - Privileged mode. Similar to user mode but with fewer restrictions.
   - Used primarily by OS kernel code.

THE 4 STATES OF OPERATION

- ARM cores have **up to four** states:
  1. **ARM64** (64-bit addressing*, registers)
  2. **ARM32** (32-bit addressing, registers)
  3. **Thumb** (16-bit instructions)
  4. **Jazelle** (8-bit Java bytecode)
- Trade-offs between speed and memory between the three states



* The embedded platform is probably not going to have 2^64 addresses, i.e. a few million terrabytes of memory… not all the lines are likely to be hooked to address lines, but it could be useful in e.g. virtual addressing, disk access, etc.

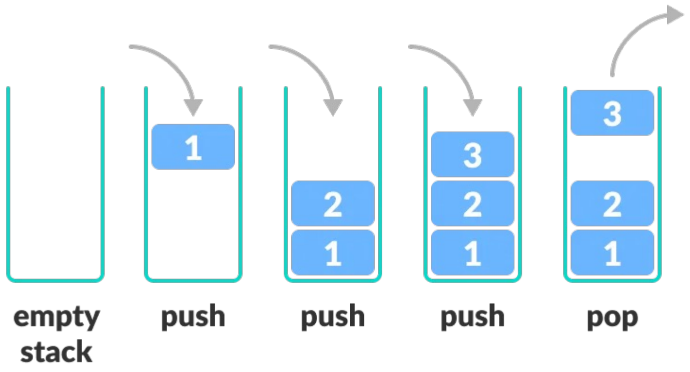
ARM REGISTERS

- ARM32 has **37** x 32-bit registers
- 1 **PC** (Program Counter)
- 1 **CPSR** (Current Program Status Register)
- 5 **SPSR** (Saved Program Status Register)
- 30 **GPR** (General Purpose Register) r0 to r14
- Only 15 GPRs are explicitly available to the programmer, the rest are used for swapping out data when changing modes (i.e. FIQ has its own registers, saves time)

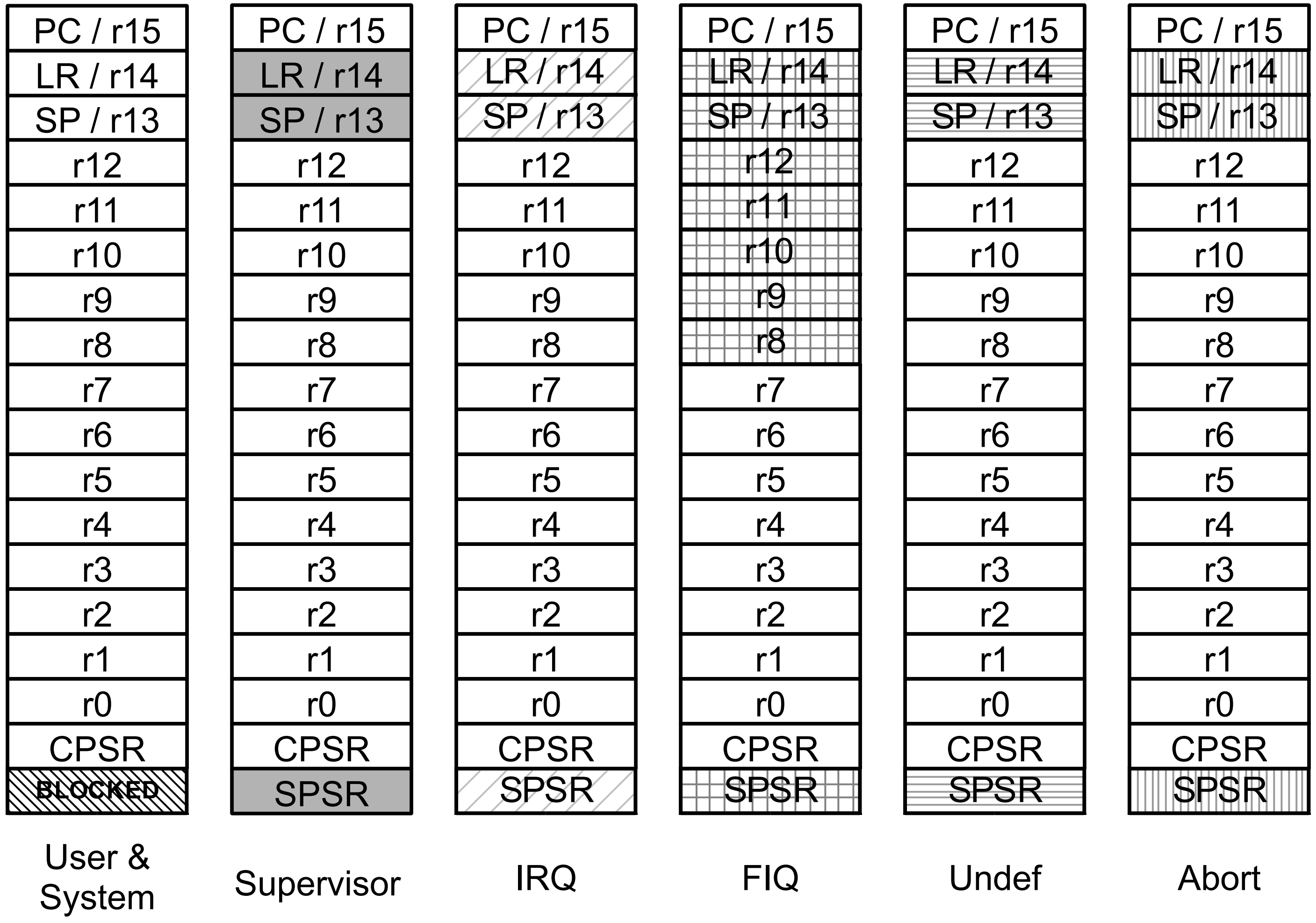
ARM REGISTERS

- Any mode can access:
  - Some general-purpose registers (r0-r12)
  - Its own **stack pointer** (r13, or sp); keeps track of position on the stack
  - Its own **link register** (r14, or lr); saves instruction address before a function call
  - The **program counter** (pc)
- All modes, except User and System, has an SPSR (Saved Program Status Register)
- The sp and lr registers can be used as GPRs
- User and System modes share sp and lr

Stack example:

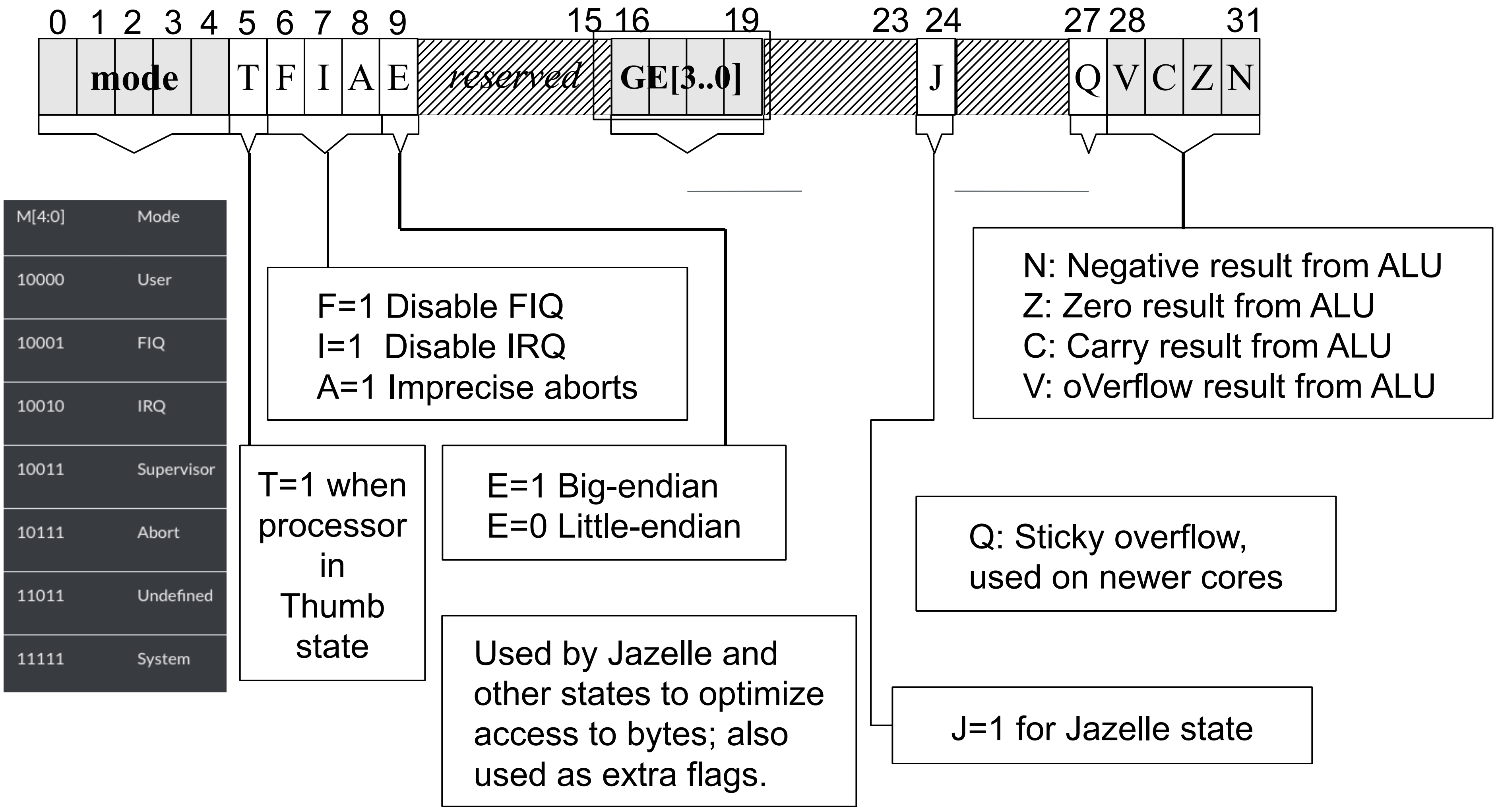


ARM REGISTER MAP



17 + 3 + 3 + 8 + 3 + 3 = **37 registers**

THE PROGRAM STATUS REGISTERS



M[4:0]	Mode
10000	User
10001	FIQ
10010	IRQ
10011	Supervisor
10111	Abort
11011	Undefined
11111	System

F=1 Disable FIQ
I=1 Disable IRQ
A=1 Imprecise aborts

T=1 when processor in Thumb state

E=1 Big-endian
E=0 Little-endian

Used by Jazelle and other states to optimize access to bytes; also used as extra flags.

N: Negative result from ALU
Z: Zero result from ALU
C: Carry result from ALU
V: oVerflow result from ALU

Q: Sticky overflow, used on newer cores

J=1 for Jazelle state

PROGRAM COUNTER REGISTER

- Called r15 or PC in most assemblers
- Use of PC depends on current state
- In ARM32 state:
  - PC values in bits 2 to 31 – bits 0 and 1 are ignored
  - Instructions are 32-bit word aligned
- In Thumb State:
  - PC values in bits 1 to 31 – bit 0 is ignored
  - Instructions are 16-bits and half-word aligned
- In Jazelle State:
  - Instructions are 8-bit, but processor reads 4 bytes (32 bits) at a time.

ARM EXCEPTION VECTOR

- An **exception** is an unexpected event from within the processor
- When an exception occurs, the processor calls a **handler** function for that exception
- The location in memory where the handler is stored is called the **exception vector**
- 32 bytes in size, starts at address 0x00000000
- Each entry point to the vector contains an ARM 32-bit instruction

FIQ	0x1C
IRQ	0x18
Reserved	0x14
Data Abort	0x10
Prefetch Abort	0x0C
Software Interrupt	0x08
Undefined Instruction	0x04
Reset	0x00

ARM EXCEPTION HANDLING

Process is as follows:

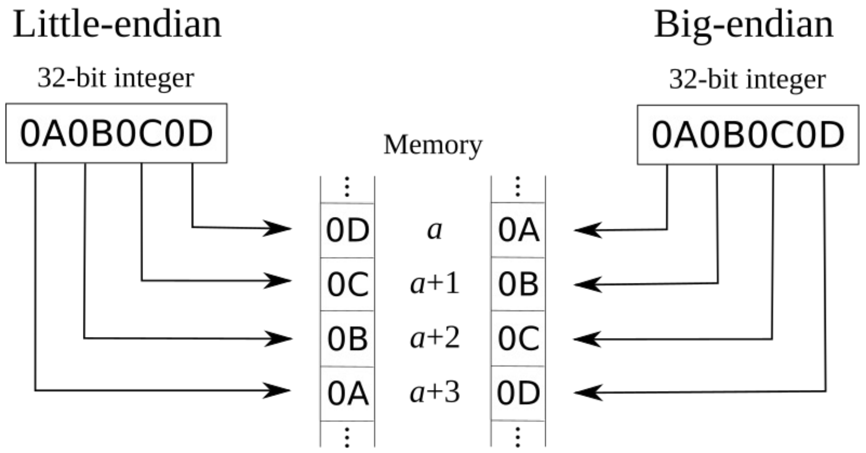
- **SPSR**[mode] = **CPSR** (save status register)
- Change to relevant exception mode, e.g. Undef mode
- Change to ARM32 state
- Disable all other Interrupts
- **LR**[mode] = **PC** (save return address)
- **PC** = address of relevant exception vector
- Resume execution (of exception handler)

RETURN FROM EXCEPTION

- Changes to ARM32 state
- **CPSR** = **SPSR**[mode] (restore status register)
- **PC** = **LR**[mode]
- Resume previous task

ENDIANNESS OF ARM

- Two ways to interpret contents of registers and memory:
- Big Endian:
  - Most significant byte first (e.g. Motorola 68000)
- Little Endian:
  - Least significant byte first (e.g. Intel x86)
- ARM supports **both** and can switch
- Can cause massive confusion…
- **My advice**: choose one and stick to it


Little-endian
32-bit integer
0A0B0C0D
Big-endian
32-bit integer
0A0B0C0D
Memory
0D
0C
0B
0A
a
a+1
a+2
a+3
0A
0B
0C
0D