# Lecture 07 (Processor Architecture)

*A2 is competitive (timewise) :')*

## 1 ARM Architecture

1. 32 bit ARM ISA
2. 16-bit Thumb ISA

## 2 ARM Operating Modes

1. User
2. FIQ - high priority interrupt is raised
3. IRQ - low priority interrupt is raised
4. Supervisor - on reset and software interrupt
5. Abort - memory access violation
6. Undef - undefined instructions
7. System - priviledge mode using same registers as user mode

## 3 ARM Register Set

1. $r_0 - r_{12}, r_{15}, oper$ is visible to all
2. $r_{13}, r_{14}$ are *sp* and *lr*

## 4 Exception Handling

### 4.1 On Exception

1. ARM copies CPSR into SPSR_<mode>
2. Sets appropriate CPSR bits
    - change to ARM state
    - change to exception mode
    - disable interrupts
3. Sets the return address in LR_<mode>

4. Sets PC value to vector address

## 4.2 On Return

1. Restores CPSR from SPSR_<mode>
2. Restores PC from LR_<mode>

# 5 Program Status Registers

## 5.1 Condition Code Flags

1. N = Negative result
2. Z = Zero result
3. C = Carried out
4. V = oVerflowed

## 5.2 Interrupt Disable Bits

1. I = 1: disables IRQ
2. F = 1: disables FIQ

## 5.3 Mode Bits

Specifies the processor mode

## 5.4 T Bit

1. Architecture xT only
2. T = 0: ARM state
3. T = 1: Thumb state

# 6 Program Counter

1. ARM state - all instructions are word aligned (and long)
2. Thumb state - all instructions are hald-word aligned (and long)
3. Jazelle state - all instructions are byte aligned (and long); 4 instructions are read at once

# 7 Conditional Execution and Flags

1. Instructions can be made to execute conditionally, instead of adding jumps
2. ADDNE == ADD if not equal
3. Flags can be set by postfixing S: SUBS == SUB and set flag
4. Conditional codes exist which set approriate flags

# 8 Data Instructions

1. Arithmetic: ADD, ADC, SUB, SBC, RSB, RSC
2. Logical: AND, ORR, EOR, BIC
3. Comparisons: CMP, TST, TEQ
4. Data movement: MOV, MVN
5. Only work on registers