# Integer Pipeline

UQ: Explain in brief integer instruction pipeline stages of Pentium

# Integer Pipeline

- The pipelines are called “u” and “v” pipes.
- The **u-pipe** can execute **any instruction**, while the **v-pipe** can execute **“simple”** instructions as defined in the “Instruction Pairing Rules”.
- When instructions are paired, the instruction issued to the **v-pipe is always the next sequential** instruction after the one **issued to u-pipe.**

# Integer Pipeline

- The integer pipeline stages are as follows:

1. **Prefetch(PF) :**
   - Instructions are prefetched **from** the on-chip **instruction cache**
2. **Decode1(D1):**
   - Two parallel decoders attempt to decode and issue the next two sequential instructions
   - It decodes the instruction to **generate a control word**

# Integer Pipeline

- A single control word causes direct execution of an instruction
- Complex instructions require microcoded control sequencing

## 3. Decode2(D2):

- **Decodes the control word**
- **Address** of memory resident **operands** are calculated

# Integer Pipeline

4. **Execute (EX):**
   - The instruction is **executed** in ALU
   - **Data** cache is **access**ed at this stage
   - For **both** ALU and data cache access requires **more than one clock.**
5. **Writeback(WB):**
   - The CPU **stores the result and updates the flags**

# Integer Instruction Pairing Rules

UQ: Explain the instruction pairing rules for Pentium

# Integer Instruction Pairing Rules

- To issue two instructions simultaneously they must satisfy the following conditions:
  - Both instructions in the pair must be **"simple".**
  - There must be **no read-after-write(RAW) or write-after-write register(WAW)** dependencies

**RAW:**

i1. **R2** ← R1 + R3
i2. R4 ← **R2** + R3

**WAW:**

i1. **R2** ← R4 + R7
i2. **R2** ← R1 + R3

# Integer Instruction Pairing Rules

- Neither instruction may contain both a **displacement** and **an immediate**
- Instruction **with prefixes** (lock,repne) can **only** occur in the **u-pipe**

# Simple Instructions

- They are entirely **hardwired**
- They do not require any microcode control
- Executes in **one clock cycle**
  - **Exception**: ALU mem,reg and ALU reg,mem are 3 and 2 clock operations respectively

- The following integer instructions are considered simple and may be paired:
  1. mov reg, reg/mem/imm
  2. mov mem, reg/imm
  3. alu reg, reg/mem/imm
  4. alu mem, reg/imm
  5. inc reg/mem
  6. dec reg/mem
  7. push reg/mem
  8. pop reg
  9. lea reg,mem
  10. jmp/call/jcc near
  11. nop
  12. test reg, reg/mem
  13. test acc, imm

# UQ: List the steps in instruction issue algorithm

# Instruction Issue Algorithm

- Decode the two consecutive instructions I1 and I2
- If the following are all true
  - I1 and I2 are simple instructions
  - I1 is not a jump instruction
  - Destination of I1 is not a source of I2
  - Destination of I1 is not a destination of I2
- Then issue I1 to u pipeline and I2 to v pipeline
- Else issue I1 to u pipeline

# Floating Point Pipeline

UQ: Explain the floating point pipeline stages.

# Floating-Point Pipeline

- The floating point pipeline has 8 stages as follows:

1. **Prefetch(PF) :**
   - Instructions are prefetched **from** the on-chip **instruction cache**
2. **Instruction Decode(D1):**
   - Two parallel decoders attempt to decode and issue the next two sequential instructions
   - It decodes the instruction to **generate a control word**

# Floating-Point Pipeline

- A single control word causes direct execution of an instruction
- Complex instructions require microcoded control sequencing

**3. Address Generate (D2):**

- **Decodes the control word**
- **Address** of memory resident **operands** are calculated

# Floating-Point Pipeline

4. **Memory and Register Read (Execution Stage) (EX):**
   - **Register read, memory read or memory write** performed as required by the instruction to access an operand.
5. **Floating Point Execution Stage 1(X1):**
   - Information from register or memory is written into FP register.
   - **Data is converted to floating point format** before being loaded into the floating point unit

# Floating-Point Pipeline

6. **Floating Point Execution Stage 2(X2):**
   - Floating point **operation performed** within floating point unit.
7. **Write FP Result (WF):**
   - Floating point results are **rounded and** the result is **written to** the **target floating point register**.
8. **Error Reporting(ER)**
   - If an error is detected, an error reporting stage is entered where the **error is reported** and **FPU status word is updated**

# Instruction Issue for Floating Point Unit

- The rules of how floating-point (FP) instructions get issued on the Pentium processor are :

1. FP instructions **do not get paired with integer instructions.**
2. When a pair of FP instructions is issued to the FPU, **only the FXCH instruction can be the second** instruction of the pair.

   The **first** instruction of the pair must be one of a set F where **F = [ FLD,FADD, FSUB, FMUL, FDIV, FCOM, FUCOM, FTST, FABS, FCHS].**

3. **FP instructions other than FXCH and instructions belonging to set F**, always get issued **singly** to the FPU.
4. FP instructions that are **not directly followed by an FXCH** instruction are issued **singly** to the FPU.