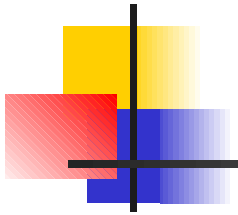
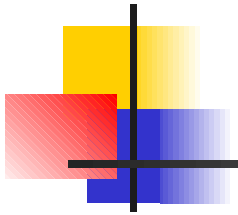


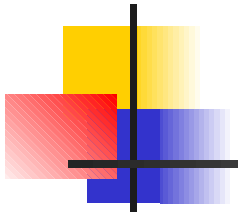
CISC vs. RISC

- **DUES FILOSOFIES DIFERENTS DE DISSENY DE COMPUTADORS**
  - RISC: Reduced Instruction Set Computer
  - CISC: Complex Instruction Set Computer



MÀQUINES CISC

- Moltes instruccions diferents
- Instruccions complexes
- Format d'instrucció variable
- Molts modes d'adreçament
- Decodificació complexa
- EXEMPLES:
  - VAX
  - Intel x86



FILOSOFIA CISC

- **EL HARDWARE ES MÉS RÀPID QUE EL SOFTWARE:**
  - **FER UN CONJUNT D'INSTRUCCIONS MOLT POTENT AMB MOLTS MODES D'ADREÇAMENT PER FER MOLTA FEINA AMB POQUES INSTRUCCIONS ASSEMBLADOR**

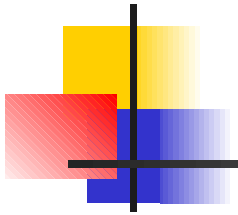
CISC - Intel 8086

```
Var
   Taula : Array [1..10] Of Byte;
   j : Byte;
Begin
   for j := 1 To 10 Do Taula[j] := 65;
End.
```

1
```
Asm
   LEA DI, Taula
   MOV AL, 65
   MOV CX, 10
@Bucle:
   MOV [DI], AL
   INC DI
   DEC CX
   JNZ @Bucle
End;
```

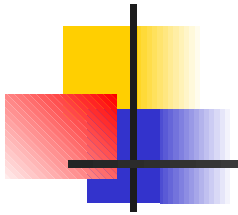
2
```
Asm
   LEA DI, Taula
   MOV AL, 65
   MOV CX, 10
@Bucle:
   STOSB
   DEC CX
   JNZ @Bucle
End;
```

3
```
Asm
   LEA DI, Taula
   MOV AL, 65
   MOV CX, 10
   REP STOSB
End;
```



MÀQUINES RISC

- Poques instruccions
- Normalment instruccions simples
- Instruccions que operen amb registres
- Format d'instrucció fixe
- Pocs modes d'adreçament
- Decodificacio simple
- EXEMPLES:
  - MIPS



FILOSOFIA RISC

- MOLT POCA GENT UTILITZA INSTRUCCIONS ASSEMBLADOR COMPLEXES
- POCA GENT UTILITZA COMPILADORS QUE GENERIN INSTRUCCIONS ASSEMBLADOR COMPLEXES

RISC - MIPS

```
Var
   Taula : Array [1..10] Of Byte;
   j : Byte;
Begin
   for j := 1 To 10 Do Taula[j] := 65;
End.
```

```
            add $6, $0, $0          # R6 := 0
            addi $6, $6, 65         # R6 := 65 (Valor a guardar)
            add $5, $0, $0          # R5 := 0
            addi $5, $5, 10         # R5 := 10 (Total iteracions)
            add $19, $0, $0         # j := 0
         Bucle:
            slt $8, $19, $5         # R8 := j >= 10
            bne $8, $0, Sortir      # Saltar si R8 != 0
            sw $6, Taula($19)       # Taula[j] = 65
            addi $19, $19, 1        # j := j + 1
            j Bucle                 # Saltar sempre a Bucle
         Sortir:
```