

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

Calcolatori — a.a. 2019–2020

Compito del 30 gennaio 2020

Cognome e Nome dello studente: _____

Un antico e glorioso algoritmo, già noto agli Egizi e riportato nel cosiddetto papiro di Rhind (compilato dallo scriba Ahmes intorno al 1650 a.C.) consente di calcolare il prodotto di due interi non negativi p e q attraverso operazioni di duplicazione e dimezzamento. L'algoritmo può essere reso in notazione moderna come segue:

```
 $r \leftarrow 0, s \leftarrow p, t \leftarrow q$  [initialize]
while  $t > 0$ :
    if  $t \% 2 \neq 0$ :  $r \leftarrow r + s$  [compute]
     $s \leftarrow 2s, t \leftarrow t // 2$  [update]
```

dove $(t//2)$ e $(t\%2)$ rappresentano rispettivamente il quoziente ed il resto della divisione intera di t per 2. Al termine dell'algoritmo, il risultato si trova in r .

Microprocessore Dato un processore CISC con un'organizzazione a singolo bus interno e bus dati e indirizzi entrambi a 16 bit, 16 registri d'uso generale a 8 bit, una ALU a 16 bit ed altra circuiteria necessaria per le operazioni di shift, disegnare la parte operativa necessaria all'esecuzione dell'istruzione `mu op1,op2,op3:op4`, dove i quattro operandi sono tutti a 8 bit. In particolare, `op1` è un operando di memoria (esprimibile in modalità diretto di memoria) o un registro, mentre i restanti operandi sono tutti registri. L'istruzione moltiplica `op1` per `op2`, ponendo il risultato in `op3` (parte più significativa) e `op4` (parte meno significativa). Esempi di sintassi: `mu R10,R11,R5:R6`, `mu var,R3,R4:R5`. Dopo aver fornito una codifica plausibile dell'istruzione, scrivere il microcodice di controllo relativo alla sua esecuzione per entrambi i modi di indirizzamento possibili per `op1`.

Programmazione LM Dato un processore RISC dotato delle stesse caratteristiche del precedente quanto a bus dati e indirizzi e a registri d'uso generale, ma con un set di istruzioni privo della `mu op1,op2,op3:op4` di cui sopra, scrivere un breve programma in linguaggio assembler che consenta comunque di eseguire l'equivalente di `mu var,R3,R4:R5` facendo uso di istruzioni più elementari. (Nota: È consentito anche scrivere il programma facendo riferimento all'architettura 8086 e servendosi delle sue istruzioni, tranne ovviamente la `mul`.)