

Soluzione appello di Architetture Innovative del 9/12/2008 (DRAFT)

L' IR è composto da:

- 2 bit di COP
- 16 bit (IR^x)
- 16 bit (IR^y)
- 16 bit (IR^z)

Quindi in totale da 50 bit. In $MAX N$, X , X è memorizzato in IR^x e N è memorizzato in IR^y .

L'aspetto fondamentale è la scelta del formato della data.

Sulla base delle istruzioni da eseguire, le date verranno rappresentate come numero di giorni trascorsi a partire da una data fissata, per es. dal 1/1/1900. Si noti che 20 bit per rappresentare le date sono sufficienti (è possibile memorizzare circa 3000 anni). Si sceglie quindi una RAM il cui MBR è di 20 bit ed il cui MAR è di 16 bit.

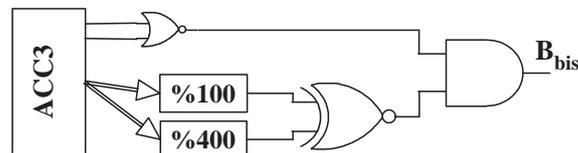
Il codice delle istruzioni richieste è riportato in seguito.

Si assume che i registri accumulatore siano a 20 bit.

La macchina è dotata di un modulo combinatorio per il riconoscimento degli anni bisestili, collegato al registro ACC3. Tale modulo produce un segnale beta, denotato con β_{bis} , che vale 1 se l'anno è bisestile, 0 altrimenti.

Il caso più semplice consiste nel considerare bisestile qualsiasi anno divisibile per 4. Con questa assunzione semplificativa il segnale β_{bis} è semplicemente $\text{NOR}(\text{ACC3}_1, \text{ACC3}_0)$.

Se si utilizza la regola più corretta: *Un anno è bisestile se è divisibile per 4, con l'eccezione che gli anni secolari (quelli divisibili per 100) sono bisestili solo se divisibili per 400*, il modulo è più complesso. Una possibile soluzione è di cablare gli anni divisibili per 100 e quelli divisibili per 400 con reti combinatorie di riconoscimento, come riportato in figura.



Ovviamente, il modulo combinatorio è sostituibile da una porzione di codice RTL che realizzano la divisione per 100 del registro ACC3.

La macchina è, inoltre, dotata di un modulo combinatorio G che riceve in input un mese m e un anno a (per gestire gli anni bisestili), e restituisce il numero di giorni che compongono il mese m dell'anno a . La sintassi per il suo utilizzo è $G[M, A] \rightarrow R$, che indica la scrittura nel registro R del numero di giorni che compongono il mese M dell'anno A .

NEXT X

```

IRx → MAR
M[MAR] → MBR
MBR → ACC1
INC(ACC1) → ACC1
ACC1 → MBR
MBR → M[MAR]

```

DIFF X, Y, Z

```

IRx → MAR
M[MAR] → MBR, IRy → MAR
M[MAR] → MBR, MBR → ACC1, IRz → MAR
MBR → ACC2
ACC1 - ACC2 → ACC3
ACC3 → MBR
MBR → M[MAR]

```

MAX N, X

```

//Ricerca del massimo
//ACC3 conserva il massimo corrente, ACC2 conserva N
IRy → ACC2, 0 → ACC3
1: if (OR(ACC2) = 1)
    IRx → MAR
    M[MAR] → MBR
    MBR → ACC1
    //Modulo compare = 1 se ACC1 > ACC3
    if (CMP(ACC1, ACC3) = 1)
        ACC1 → ACC3, INC(IRx) → IRx, DEC(ACC2) → ACC2, goto 1
    else
        INC(IRx) → IRx, DEC(ACC2) → ACC2, goto 1
    fi
else
    //Coneversione di una data nel formato scelto in (Giorno, Mese, Anno)
    //R contiene il residuo dei giorni, A e B registri tampone della ALU
    ACC3 → R, '1899' → ACC3, 0 → ACC2, 0 → ACC1, 365 → B
    2: if (R31 == 0) then //finché R ≥ 0
        R → A, INC(ACC3) → ACC3
        if (βbis = 1) then //l'anno è bisestile
            A - B - 1 → R, goto 2
        else
            A - B → R, goto 2
        fi
    else //In ACC3 ho l'anno corretto, cerco il mese
        A → R
        3: if (R31 == 0) then //finché R ≥ 0
            INC(ACC2) → ACC2
            R → A, G[ACC2, ACC3] → B
            A - B → R, goto 3
        else //In ACC2 ho il mese corretto, cerco il giorno
            A → ACC1 + 1
        fi
    fi
fi

```