

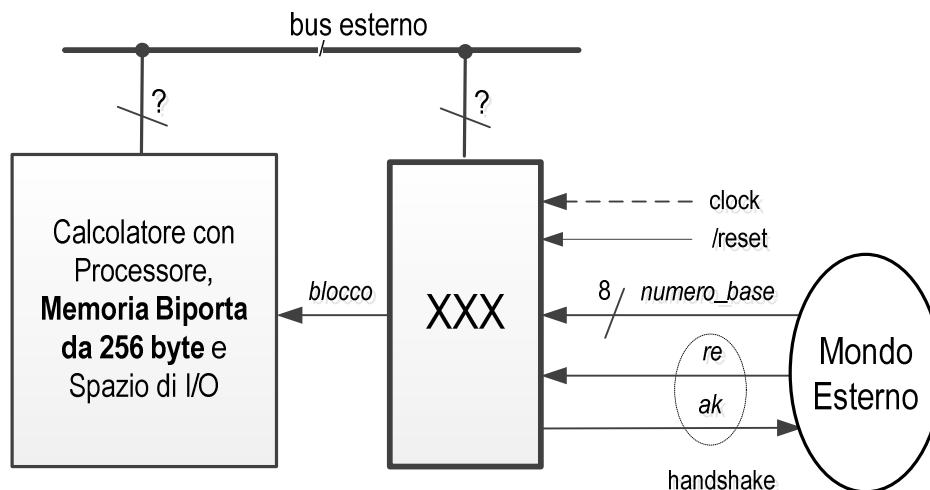
Nome e Cognome: \_\_\_\_\_ Matricola: \_\_\_\_\_

### Esercizio 1

Descrivere una RSS di Moore, e sintetizzarla secondo il modello con D-FF come elementi di marcatura. La rete ha un ingresso ed un'uscita, e *ritarda di 3 clock l'ingresso*.

Si consiglia di svolgere la descrizione partendo da uno stato iniziale in cui la rete ha ricevuto 000 negli ultimi tre clock, e produce zero in uscita.

### Esercizio 2



**PREMESSA:** Quando la variabile *blocco* vale 1, il processore si blocca in un tempo non superiore a un ciclo e mezzo di clock e la **memoria biporta** diviene gestibile tramite il **bus esterno**.

**Descrivere l'Unità XXX** in accordo alle seguenti specifiche:

- Quando la variabile di ingresso *re* vale 0, l'Unità XXX è in una condizione di riposo in cui tiene a 0 le variabili di uscita *ak* e *blocco*
- Quando la variabile di ingresso *re* va ad 1, l'Unità XXX passa in una condizione di lavoro in cui preleva il *numero\_base*, mette a 1 le variabili di uscita *ak* e *blocco* ed entra in una fase in cui scrive nella memoria, a partire dall'indirizzo 00H, il *numero\_base* e i 255 numeri ad esso successivi (si indichi come A7\_A0 il registro che supporta gli indirizzi).
- Quando questa operazione è compiuta, l'Unità XXX mette la variabile *blocco* a 0 e passa a testare la variabile di ingresso *re*.
- Quando la variabile *re* è di nuovo a 0, l'Unità XXX torna nella condizione di riposo mettendo la variabile *ak* a 0 e così via all'infinito.

**Sintetizzare l'Unità XX** disegnando circuitalmente la parte operativa relativa al registro A7\_A0.

NOTA: Si supponga che la memoria sia sufficientemente veloce da non richiedere stati di wait quando viene acceduta

### Esercizio 1 – una possibile soluzione

Conviene codificare lo stato come la sequenza degli ingressi visti negli ultimi 3 clock:  $S_{000}, S_{001}, \dots, S_{111}$ . Con questa premessa, l'uscita è il bit meno significativo del nome dello stato, e la tabella di flusso è la seguente:

		x		z
		0	1	
S	$S_{000}$	$S_{000}$	$S_{100}$	0
	$S_{001}$	$S_{000}$	$S_{100}$	1
	$S_{010}$	$S_{001}$	$S_{101}$	0
	$S_{011}$	$S_{001}$	$S_{111}$	1
	$S_{100}$	$S_{010}$	$S_{110}$	0
	$S_{101}$	$S_{010}$	$S_{110}$	1
	$S_{110}$	$S_{011}$	$S_{111}$	0
	$S_{111}$	$S_{011}$	$S_{111}$	1

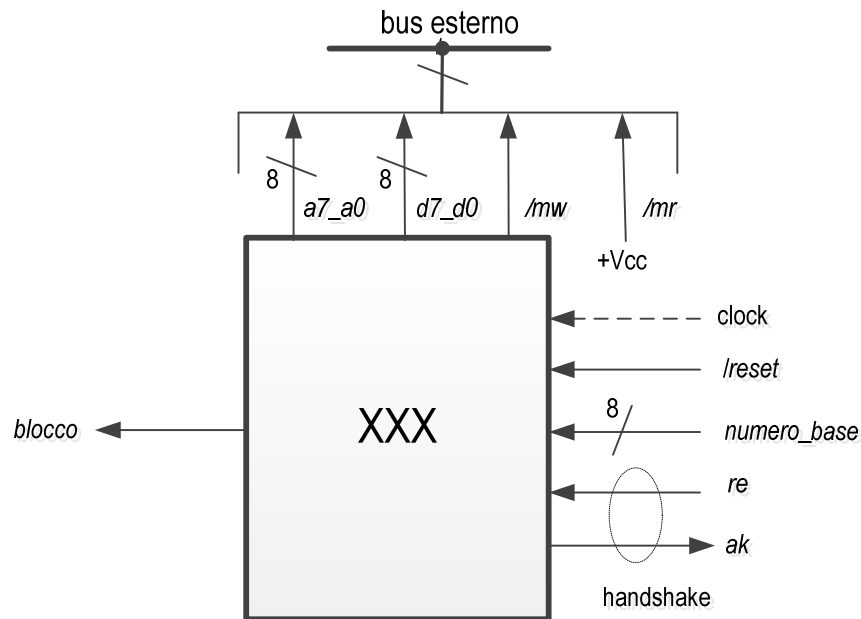
La rete RC\_B è un corto circuito ( $z = y_0$ ), mentre per la rete RC\_A si ottiene la seguente tabella:

		xy0			
		00	01	11	10
y2y1	00	000	000	100	100
	01	001	001	101	101
	11	011	011	111	111
	10	010	010	110	110

a2a1a0

Dalla quale si ottiene immediatamente:  $a_2 = x$ ,  $a_1 = y_2$ ,  $a_0 = y_1$ , come era lecito attendersi.

Nome e Cognome: \_\_\_\_\_ Matricola: \_\_\_\_\_

**Es 2 – Una possibile soluzione**

```

module XXX(a7_a0,d7_d0,mw_,blocco, numero_base,re,ak, clock,reset_);
    input          clock,reset_;
    // Connessioni di XXX con il computer
    output [7:0] a7_a0;
    output [7:0] d7_d0;
    output      mw_;           //mr_ e' stato cablato fisso ad 1
    output      blocco;

    // Connessioni con il mondo esterno
    input [7:0]  numero_base;
    input       re;
    output      ak;

    //Registri
    reg [7:0] A7_A0,D7_D0; assign a7_a0=A7_A0; assign d7_d0=D7_D0;
    reg MW_,AK,BLOCCO;      assign mw_=MW_; assign ak=AK; assign blocco=BLOCCO;
    reg [2:0] STAR;         parameter S0=0,S1=1,S2=2, S3=3,S4=4,S5=5,S6=6;

    // CICLO
    always @(reset_==0) begin MW_<=1; BLOCCO<=0; AK<=0; STAR=S0; end
    always @(posedge clock) if (reset_==1) #3
        casex(STAR)
            S0: begin AK<=0; D7_D0<=numero_base; A7_A0<='H00;
                  STAR<=(re==0)?S0:S1; end
            S1: begin BLOCCO<=1; AK<=1; STAR<=S2; end
            S2: begin STAR<=S3; end
            S3: begin MW_<=0;STAR<=S4; end
            S4: begin MW_<=1;STAR<=S5; end
            S5: begin A7_A0<=A7_A0+1; D7_D0<=D7_D0+1; STAR<=(A7_A0==255)?S6:S3; end
            S6: begin BLOCCO<=0; STAR<=(re==1)?S6:S0; end
        endcase
endmodule

```