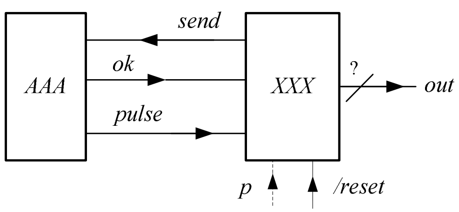
Esercizio 1

Si consideri un piano cartesiano a coordinate intere, rappresentate su *n* bit in complemento alla radice. Siano *a, b, c,* i tre coefficienti di una parabola e sia *x* un'ascissa sul piano. Si assuma che tutti i numeri sopra elencati siano rappresentabili.

1) Sintetizzare la rete combinatoria che prende in ingresso le rappresentazioni *A, B, C, X* dei 4 numeri *a, b, c, x* e produce in uscita su ? bit la rappresentazione *M* del numero *m*, coefficiente angolare della tangente alla parabola nel punto di ascissa *x*.
2) Assumendo *n*>2, descrivere tramite mappa di Karnaugh (senza sintetizzare) la rete combinatoria che produce i due bit meno significativi di *M*, m_1m_0. Si consiglia di seguire i seguenti passi:
   a. Individuare da quali variabili logiche dipendono m_1m_0
   b. risalire all'indietro da queste fino agli ingressi che le producono.
   c. Scrivere la mappa di Karnaugh.

Esercizio 2



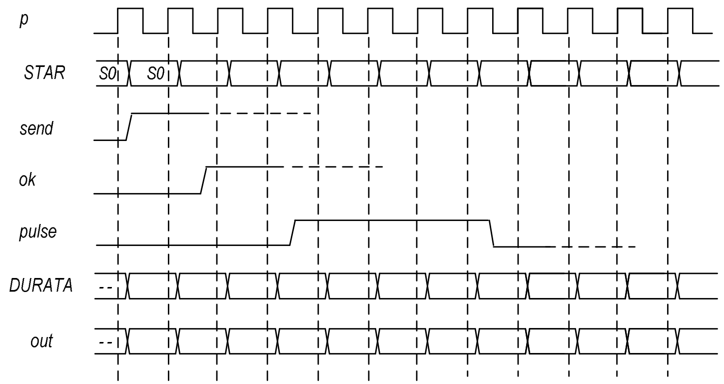
L'Unità *XXX* inizia un handshake con il circuito *AAA* settando *send*. Il circuito *AAA* risponde settando *ok* e, dopo un tempo non inferiore ad un periodo del clock *p* (questa è una ipotesi semplificativa), invia un impulso all'unità *XXX* tramite la variabile *pulse*. L'unità *XXX* calcola la durata dell'impulso (misurata come numero dei fronti in salita del clock *p* che vedono *pulse* a 1) e, quando l'impulso è terminato, resetta *send*, presenta la durata dell'impulso tramite *out*, attende che *AAA* resetti *ok* e quindi torna a gestire un nuovo handshake con *AAA*.

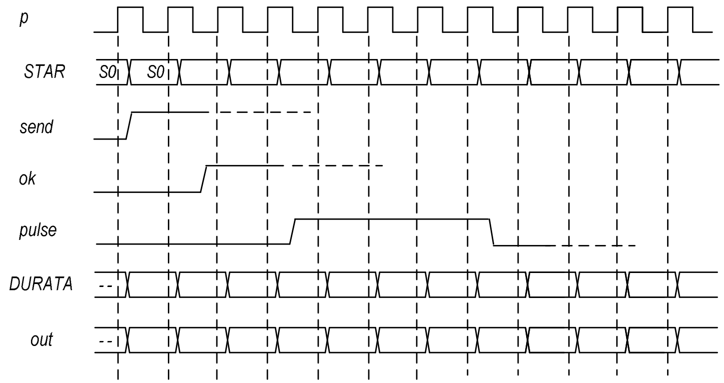
Al reset l'Unità *XXX* pone *send* a 0 e similmente il circuito *AAA* pone *ok* e *pulse* a 0. La variabile *out* è modificata dall'Unità *XXX* esclusivamente in concomitanza con la presentazione della durata di un nuovo impulso.

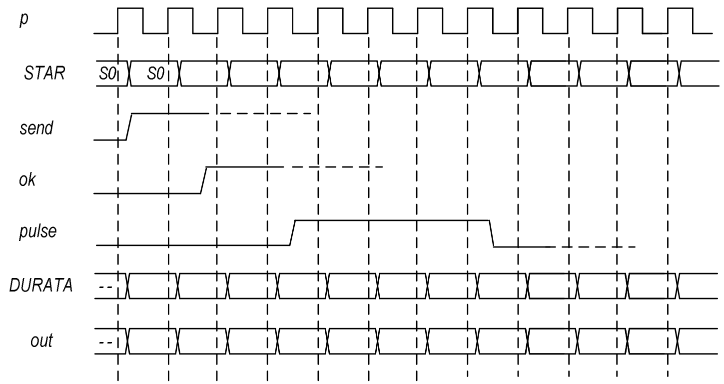
Altra ipotesi semplificativa:

b) La durata di un impulso, misurata come detto sopra, non è necessariamente uguale da impulso a impulso, ma è comunque compresa fra **3** e **62**.

***Descrivere* l'Unità *XXX* e *disegnarne* l'evoluzione completando il diagramma allegato**

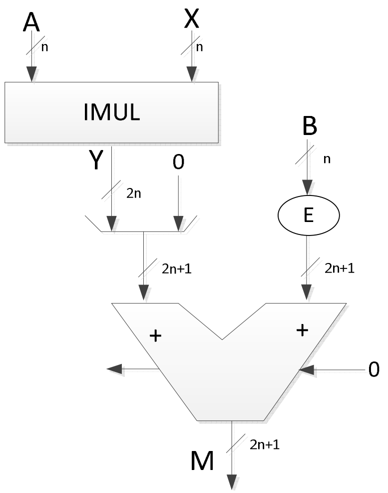






Es 1 - Soluzione

1) Il risultato da ottenere è il seguente: $m = ax + b$. Il numero minimo di bit richiesto per M è 2n+1, come si può verificare meccanicamente con un calcolo semplice. La rete che produce il risultato è quella a destra (che non sente l'ingresso C).



2) Come si vede dalla figura, m_0 dipende *soltanto* dal bit meno significativo dei due ingressi del sommatore. Uno di questi due bit vale zero, quindi $m_0 = b_0$. Per quanto riguarda m_1, questo dipende da b_1 e dal bit meno significativo di Y, y_0, che vale $y_0 = a_0 \cdot x_0$. Quindi, $m_1 = b_1 + a_0 \cdot x_0$.

La rete combinatoria richiesta ha quindi quattro ingressi, a_0, x_0, b_1, b_0, e due uscite, m_1, m_0, ed è la seguente:

m_1m_0

b_1b_0 \ a_0x_0	00	01	11	10
00	00	00	10	00
01	01	01	11	01
11	11	11	01	11
10	10	10	00	10

Es 2 - Una Possibile Soluzione

```
module XXX(send,ok,pulse, out, p,reset_);
 input         p,reset_;
 input         ok,pulse;
 output        send;
 output [5:0]  out;

 reg        SEND;  assign send=SEND;
 reg [5:0]  DURATA, OUT;  assign out=OUT;

 reg [1:0]  STAR;
 parameter  S0=0,S1=1,S2=2,S3=3;


 always @(posedge p or negedge reset_)
  if (reset_==0) begin SEND=0; STAR=S0; end
  else #3
   casex(STAR)
     S0:   begin SEND<=1; DURATA<=0; STAR<=(ok==0)?S0:S1; end
     S1:   begin DURATA<=DURATA+pulse; STAR<=(pulse==0)?S1:S2; end
     S2:   begin DURATA<=DURATA+pulse; STAR<=(pulse==1)?S2:S3; end
     S3:   begin SEND<=0; OUT<=DURATA; STAR<=(ok==1)?S3:S0; end
   endcase
endmodule
```