Approssimazione Lineare di Funzioni in un Contesto Rispettoso della Privacy

Tommaso Pignata

Contenuti della presentazione

- Obbiettivo
- Introduzione ai Garbled Circuit
- Funzione nello spazio cifrato
  - Descrizione dei Vincoli
  - Rappresentazione dei Dati
- Costruzione della Spezzata
- Schema del Circuito
- Conclusioni

Obbiettivo

-Approssimare una funzione non lineare nel dominio cifrato
- Garbled Circuit
- Ibrido (Garbled Circuit e Omomorfia)

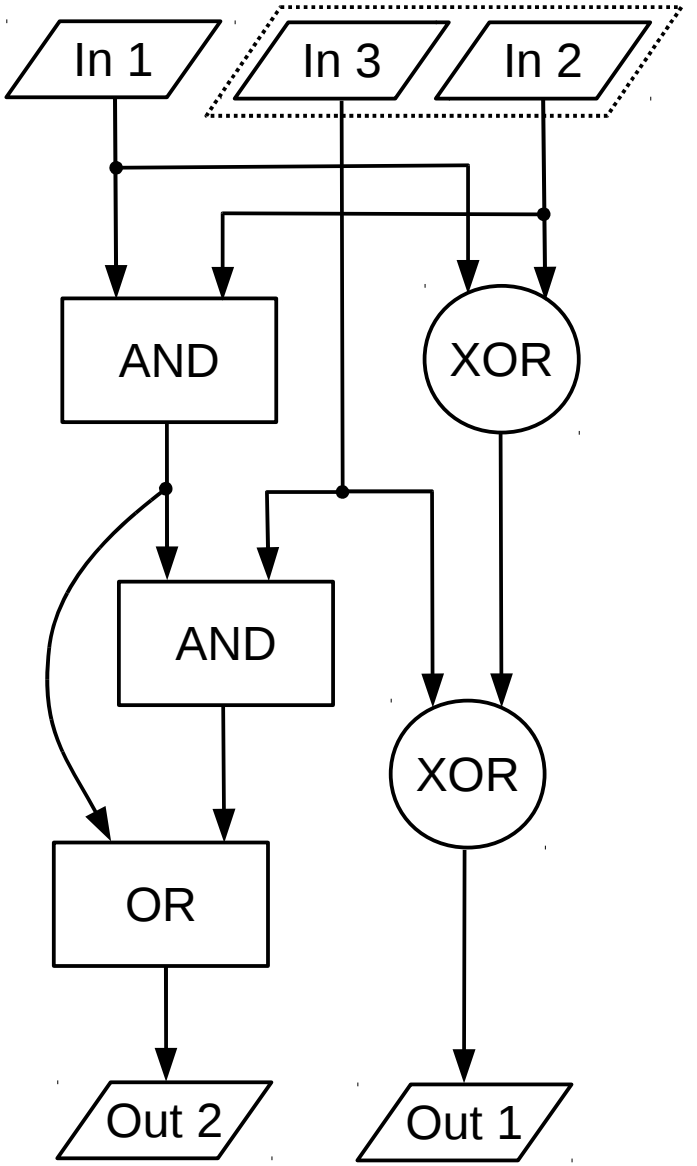
-Applicazioni:
- Funzioni di attivazione di reti neurali (arcotangente)
- Molto altro

Introduzione ai Garbled Circuit /1

Sono circuiti binari

- I/O hanno un numero definito di bit
- grafo in cui ogni nodo è una porta logica
- implementano qualsiasi operazione

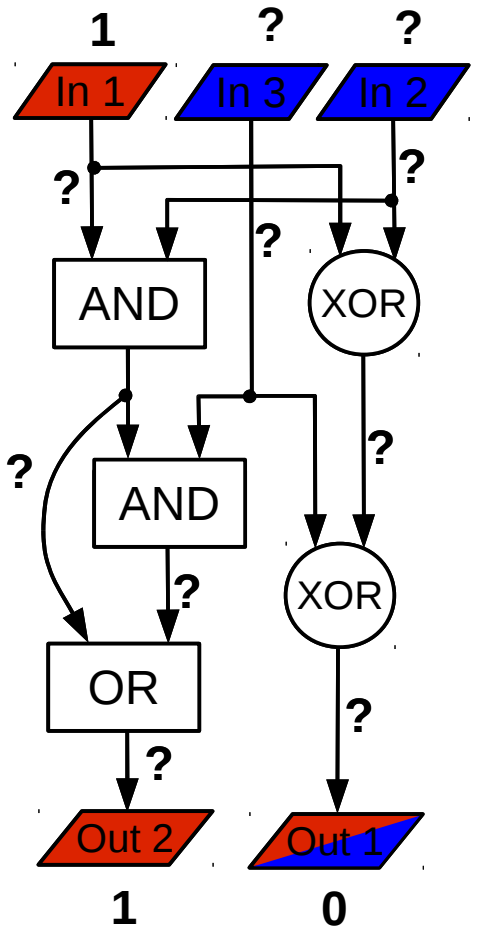
Si implementano attraverso protocolli crittografici anzichè componenti elettroniche



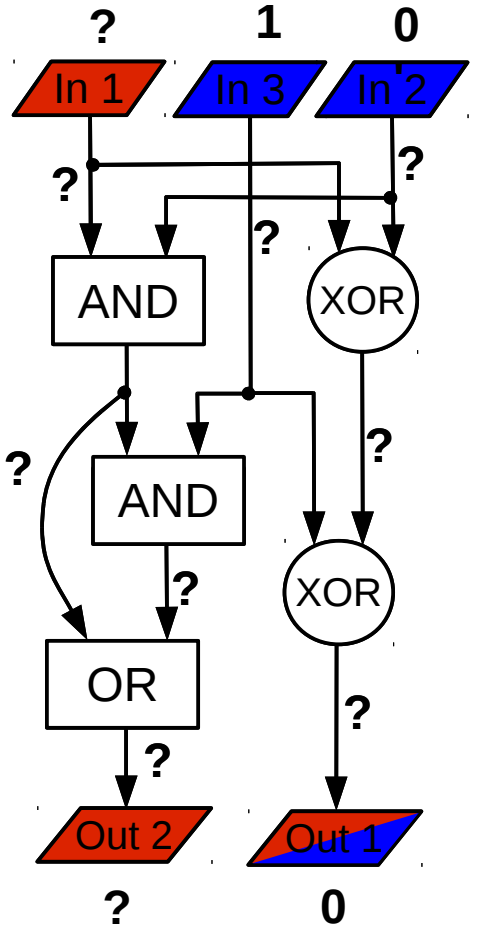
Introduzione ai Garbled Circuit /2

Calcolo collaborativo

Quello che vede *Alice*



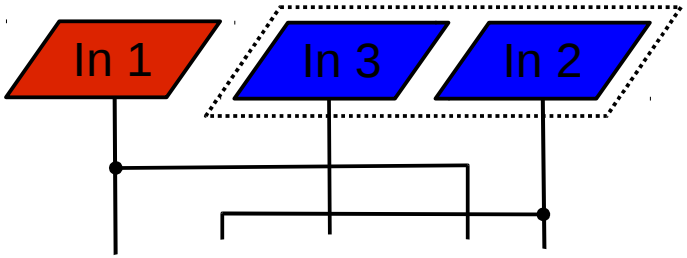
Quello che vede *Bob*



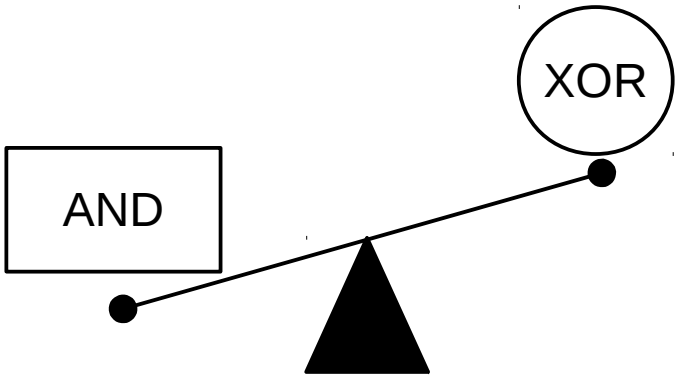
Introduzione ai Garbled Circuit /3

Caratteristiche /1

- I/O appartengono ad *Alice* o *Bob*

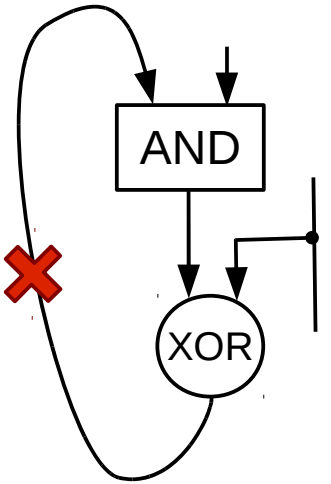


- complessità del protocollo dipende dal numero di porte (escluse XOR, NOT e NXOR)



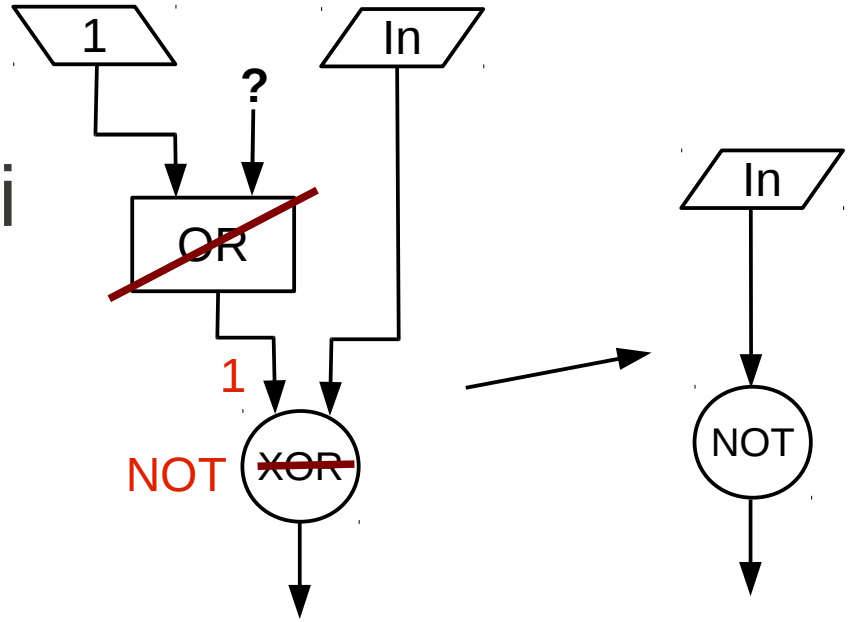
Introduzione ai Garbled Circuit /4

Caratteristiche/2



- niente feedback (il circuito è un grafo è aciclico)

- conviene avere molti valori costanti (si riduce il numero di porte)



Introduzione ai Garbled Circuit /5

Funzionamento

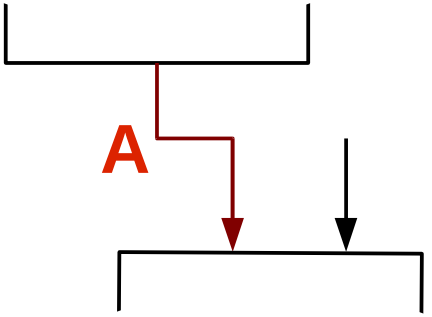
1. Alice associa ad ognuno dei due possibili valori di una linea un segreto casuale (di 80 bit).
2. Bob conosce solo uno dei due segreti (corrispondente al corrente valore della linea).

Alice

A

`0xAE45E19D` → 0

`0x10A45C11` → 1



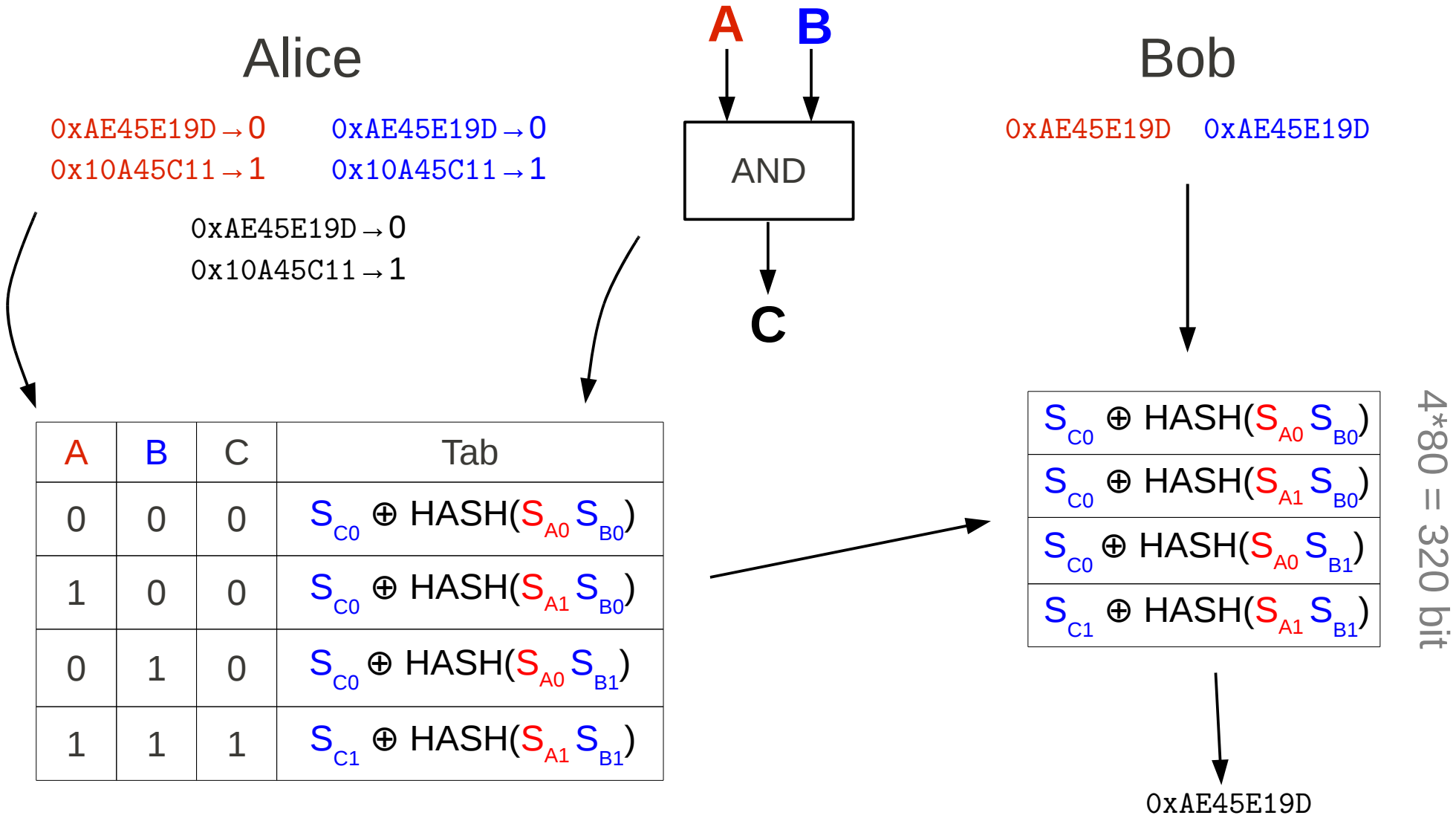
Bob

A

`0xAE45E19D`

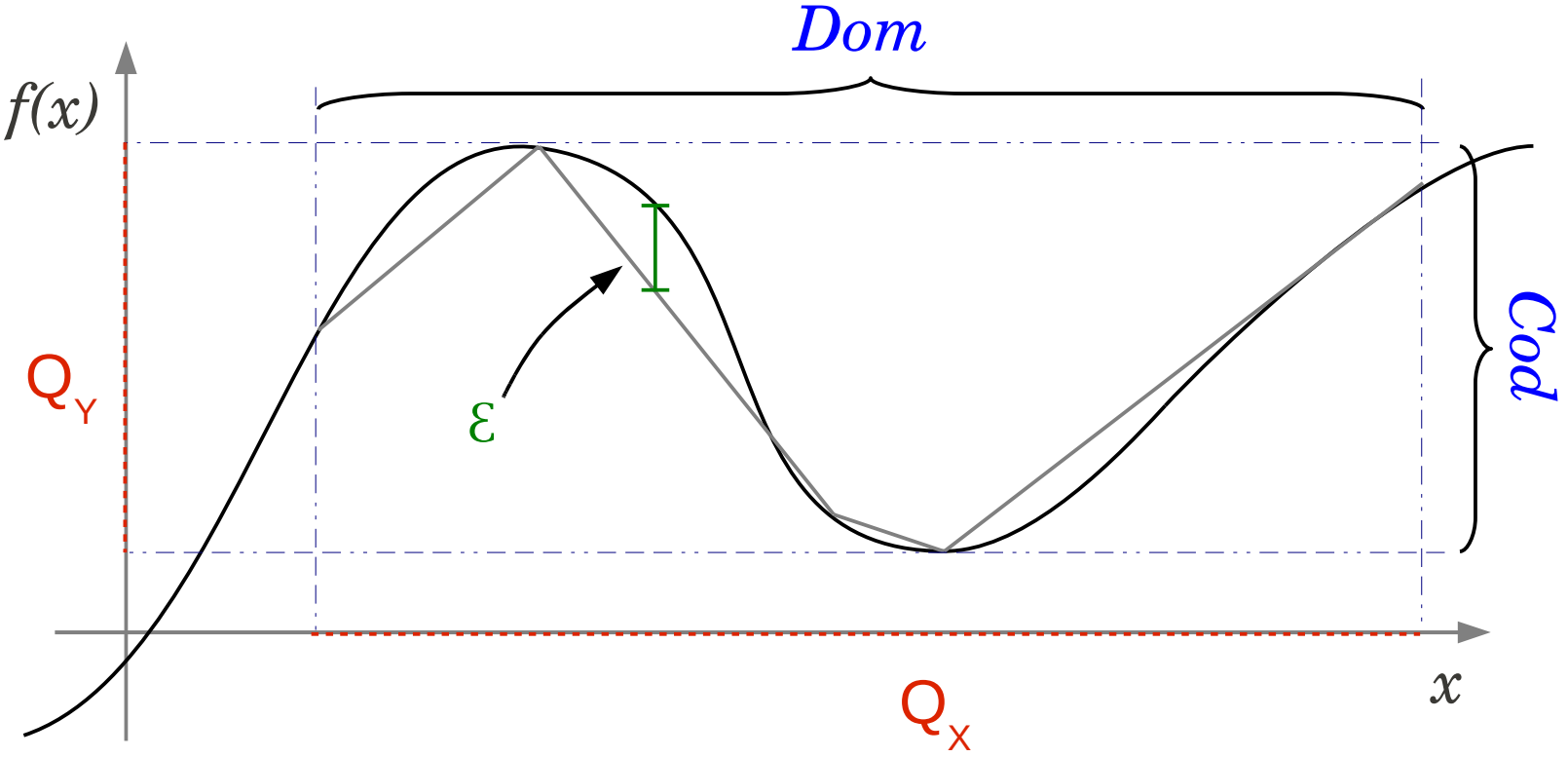
Introduzione ai Garbled Circuit /6

Esempio (AND)



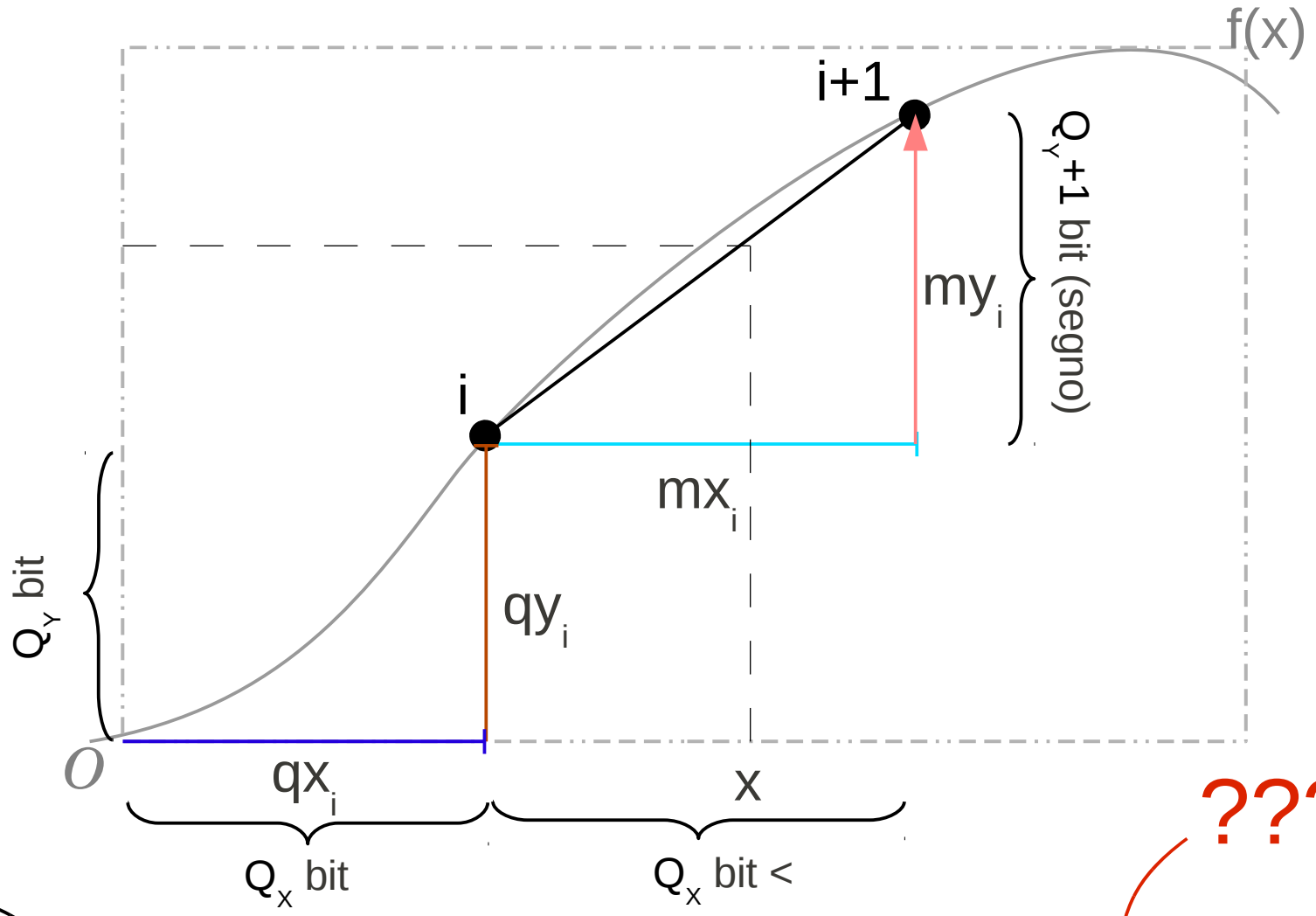
Funzione nello spazio cifrato /1
Vincoli

- Lavoriamo in uno spazio finito (Dom x Cod)
- Approssimiamo la funzione mediante una spezzata di N segmenti (a meno di un errore fissato ε)
- Valori nel Dom e Cod vengono quantizzati e rappresentati con Q_X e Q_Y bits



Funzione nello spazio cifrato /2

Parametri di un Segmento



???

Q_Y

$f(x) \simeq (x-qx_i)*m_i+qy_i$

Q_X Q_Y

dove $m_i = (my_i/mx_i)$

Q_Y Q_X

Funzione nello spazio cifrato /3

Rappresentazione Intera di Coefficienti Angolari

Coefficiente Angolare Reale: $mr_i = \frac{my_i}{mx_i} \notin Z$

Coefficiente Angolare Intero: $m_i = \frac{my_i * mcm}{mx_i} \in Z$

dove $mcm \equiv MCM(\{mx\}_{i=0,1,2...})$

Bit necessari per la rappresentazione:

$$\lceil \log_2(MAX(\{my_i\}_{i=0,1,2...} * mcm) \rceil =$$

$$\lceil \log_2 (MAX(\{my_i\}_{i=0,1,2...}) + \log_2 (mcm) \rceil$$

Problema :

mcm può essere molto grande (molti numeri primi differenti)

Funzione nello spazio cifrato /4

Rappresentazione Intera di Coefficienti Angolari

Soluzione:

Imponiamo $mx_i \equiv 2^{ai}$ per i = 0,1,2...

$$mcm \equiv MCM(\{mx\}_{i\ i=0,1,2...}) = MAX(\{mx\}_{i\ i=0,1,2...})$$

Bit necessari per la rappresentazione di mcm:

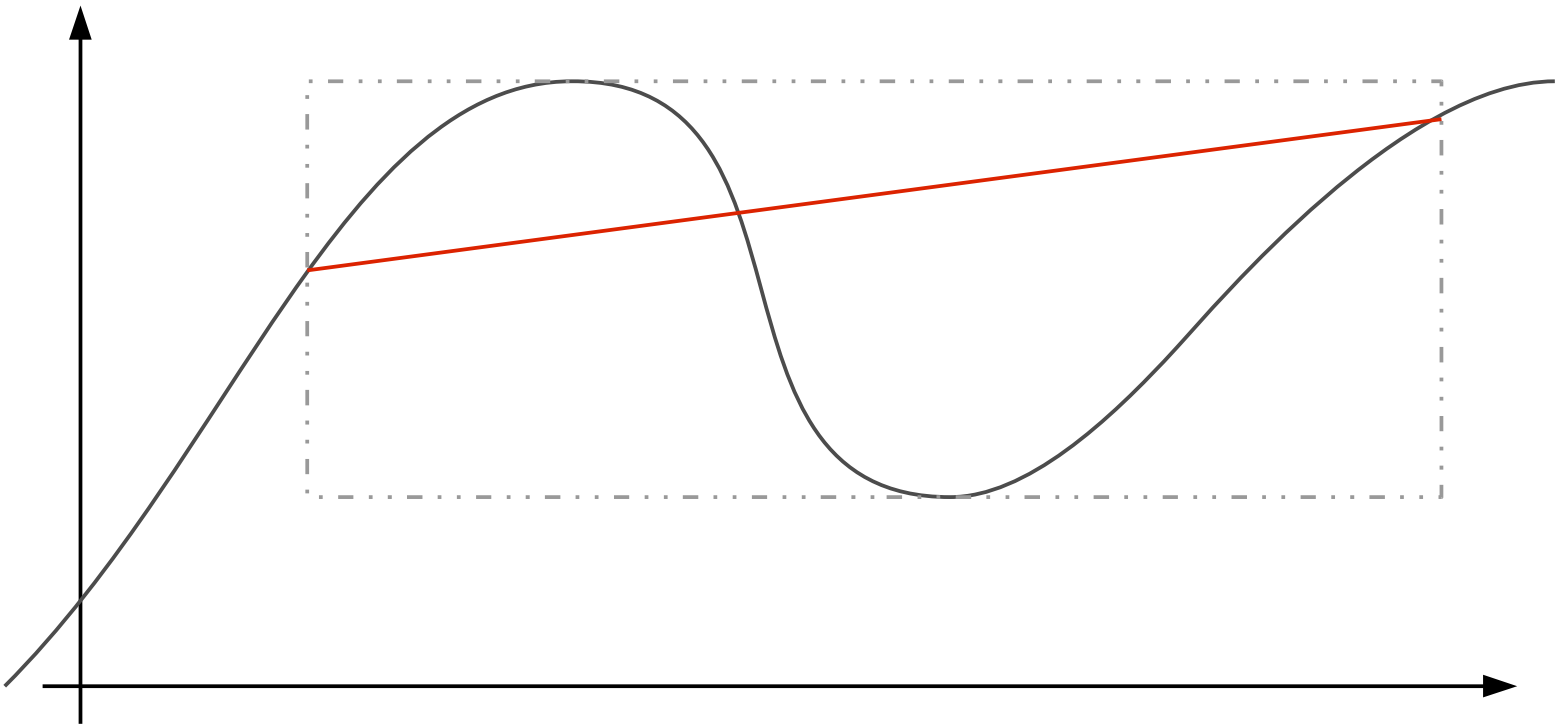
$$mcm = MAX(\{a\}_{i\ i=0,1,2...})+1$$

Bit necessari per la rappresentazione di m :

$$M = \lceil \log_2(MAX(\{my\}_{i\ i=0,1,2...}) + \log_2(mcm) \rceil = Q_Y + Q_{mcm}$$

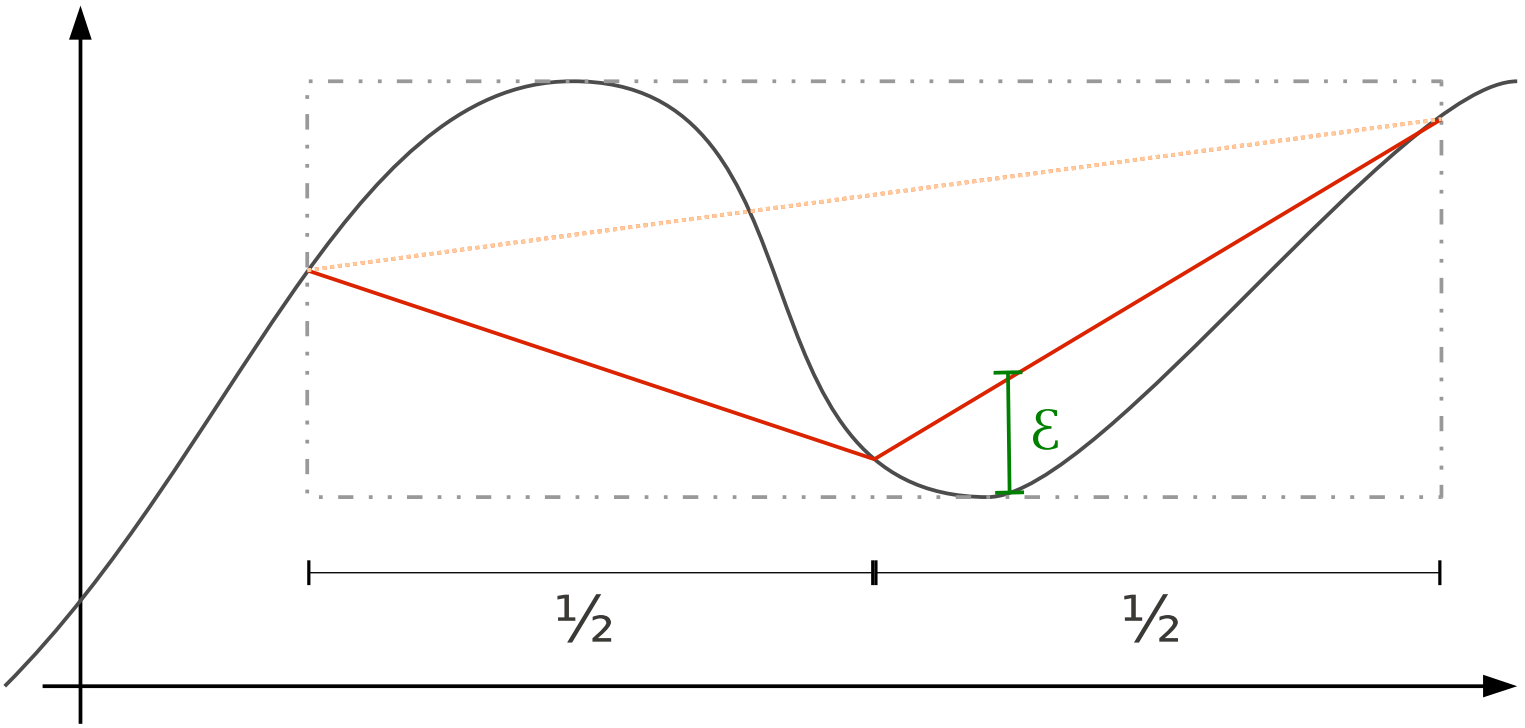
Costruzione della Spezzata per Bisezione /1

1

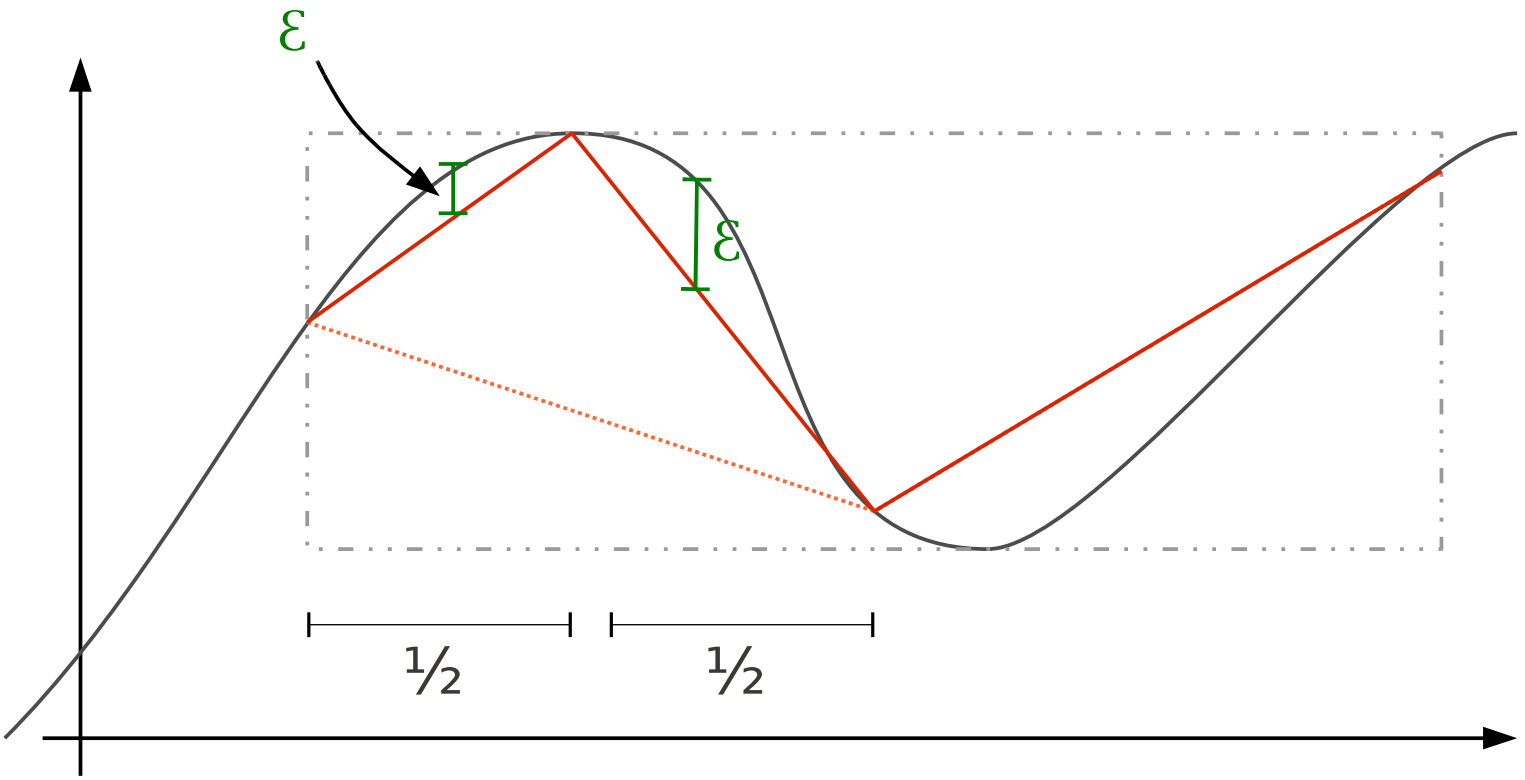


Costruzione della Spezzata per Bisezione /2

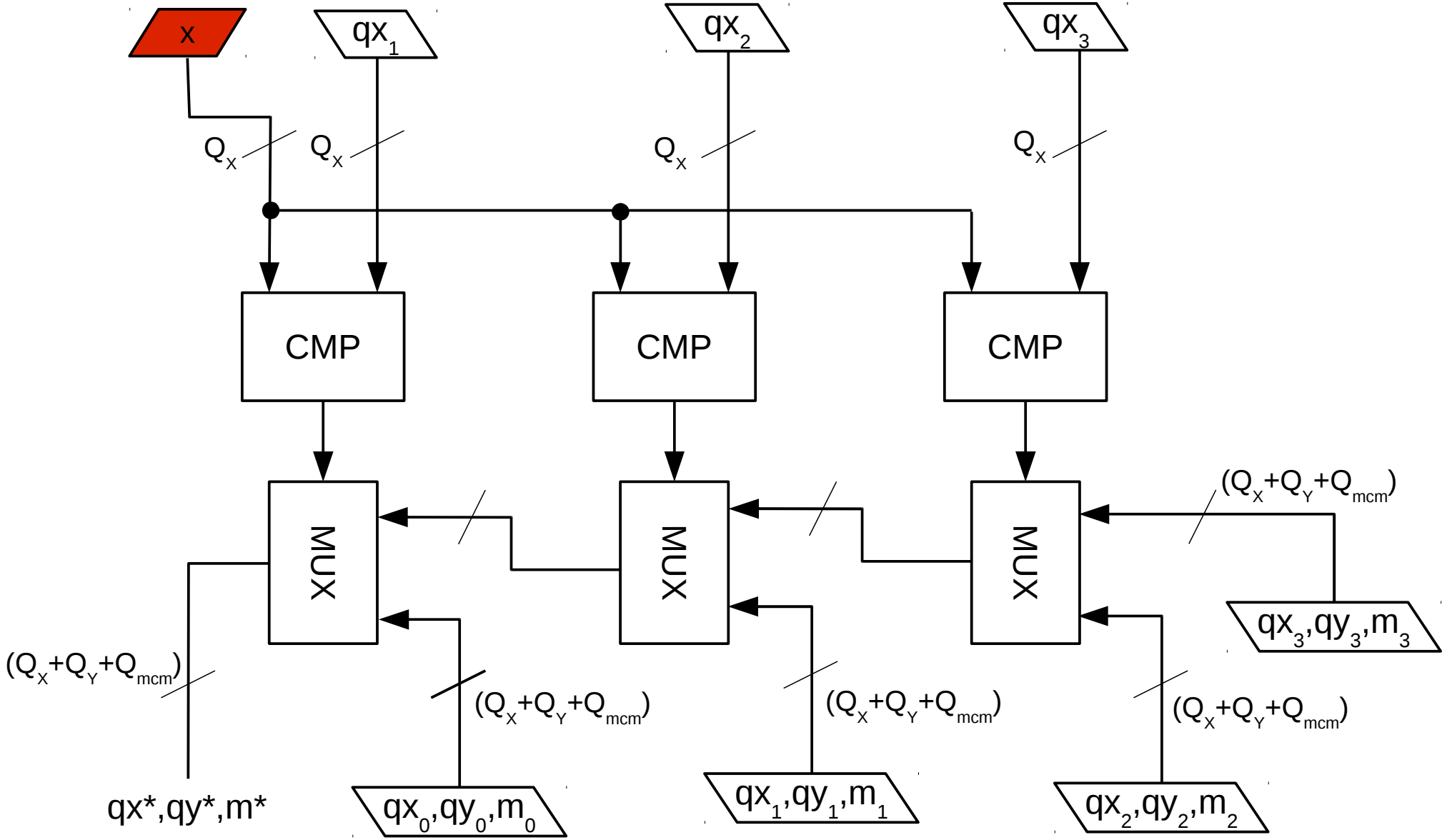
2



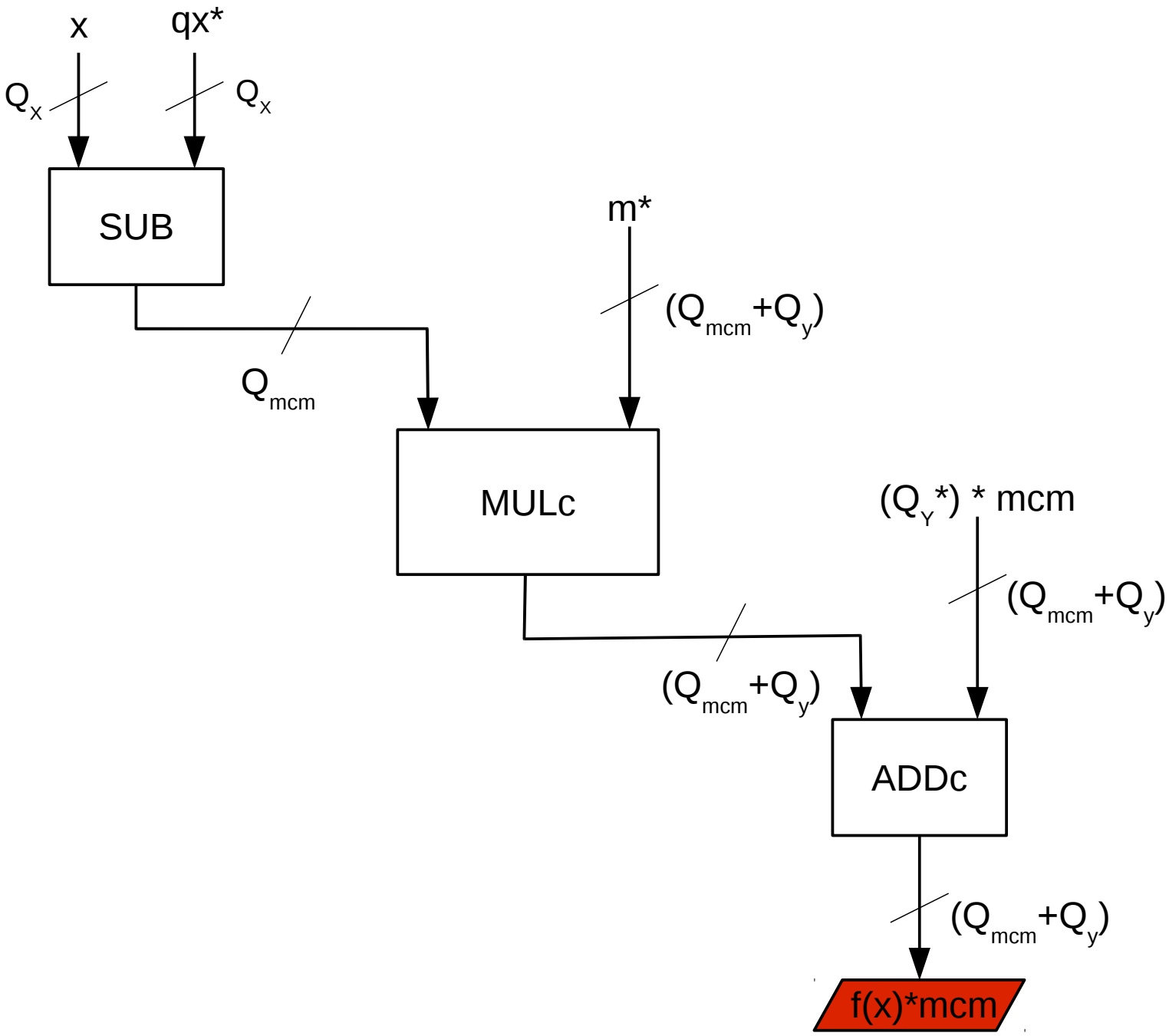
Costruzione della Spezzata per Bisezione /3



Schema del Circuito /1



Schema del Circuito /2



Schema del Circuito /3
Numero di porte totali

Sottocircuito	# di porte (escluse XOR, NOT)
Comparatori	$(N-1)Q_X$
Multiplexer	$(N-1)Q_X + 2Q_Y + Q_{mcm}$
Sottratori	Q_{mcm}
Sommatori	$Q_{mcm} + Q_Y$
Moltiplicatori	$2Q_{mcm}\,(Q_Y - 1)$
Totale	$2(N-1)Q_X + 3Q_Y + Q_{mcm} + 2Q_{mcm}\,Q_Y$

La tabella non tiene conto dell'eliminazione delle tabelle data dagli ingressi costanti.
Sperimentalmente si osserva una riduzione di circa di il 50%(~0.56).

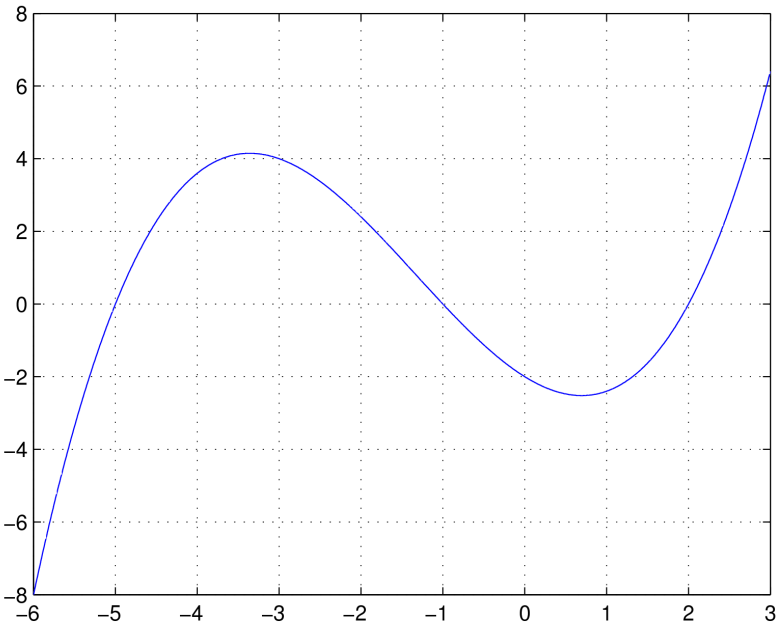
$$\#\text{ di porte non XOR oNOT} = (N-1)Q_X + (3Q_Y + Q_{mcm})/2 + Q_{mcm}\,Q_Y$$

Conclusioni /1
Dati Sperimentali /1

Prove di 'segmentazione' su un polinomio generico

$$p(x) = (x^3+4x^2-7x-10)/5$$

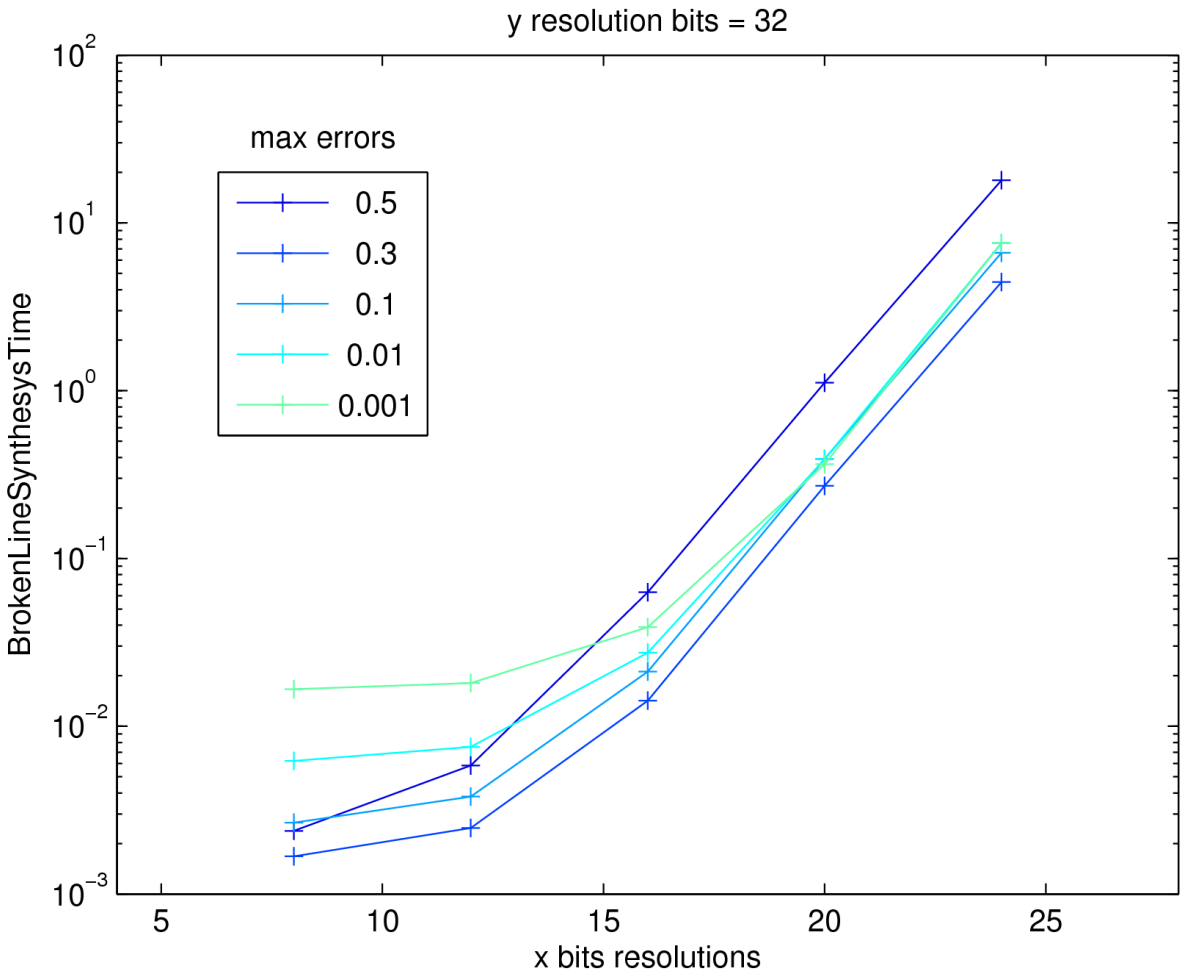
- $\varepsilon \in \{0.5, 0.3, 0.1, 0.01, 0.001\}$
- Dom [-6,+3)



Contenuto	Costante	Intervallo di Valori
Bit di quantizzazione dell'asse x	QX	8 − 24
Bit di quantizzazione dell'asse y	QY	8 − 24
Bit di quantizzazione per l'mcm	Qmcm	4 − 12
Numero di segmenti	N	10 − 210

Conclusioni /2
Dati Sperimentali /2

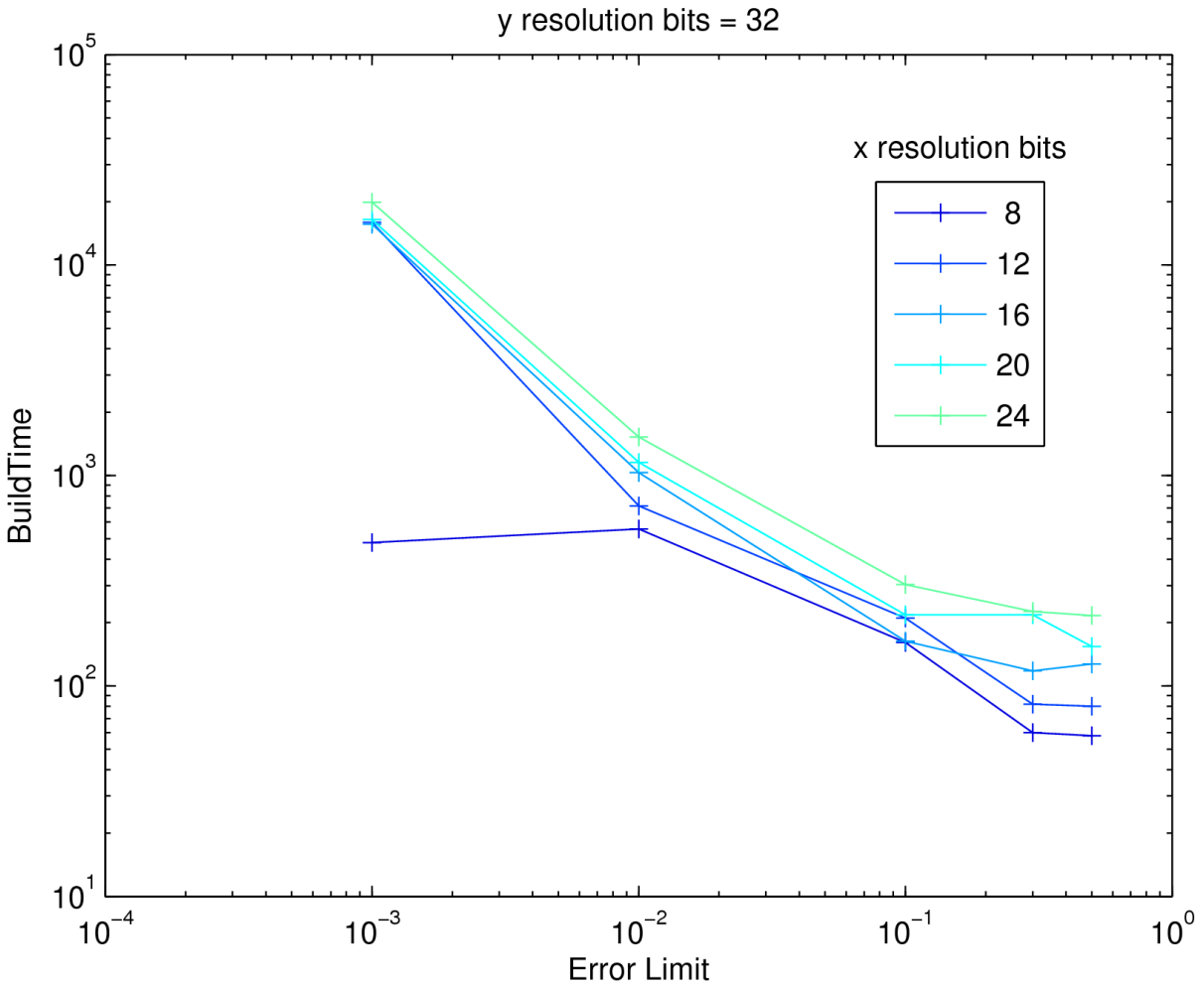
Tempo di ricerca dell'approssimazione (Matlab)



- Dipende in modo trascurabile dal numero di bit Q_Y utilizzati per rappresentare l'uscita.
- Dipende poco dal livello di precisione ricercato ε
- Dipende molto dal numero di bit Q_X utilizzati per rappresentare l'ingresso.

Conclusioni /3
Dati Sperimentali /3

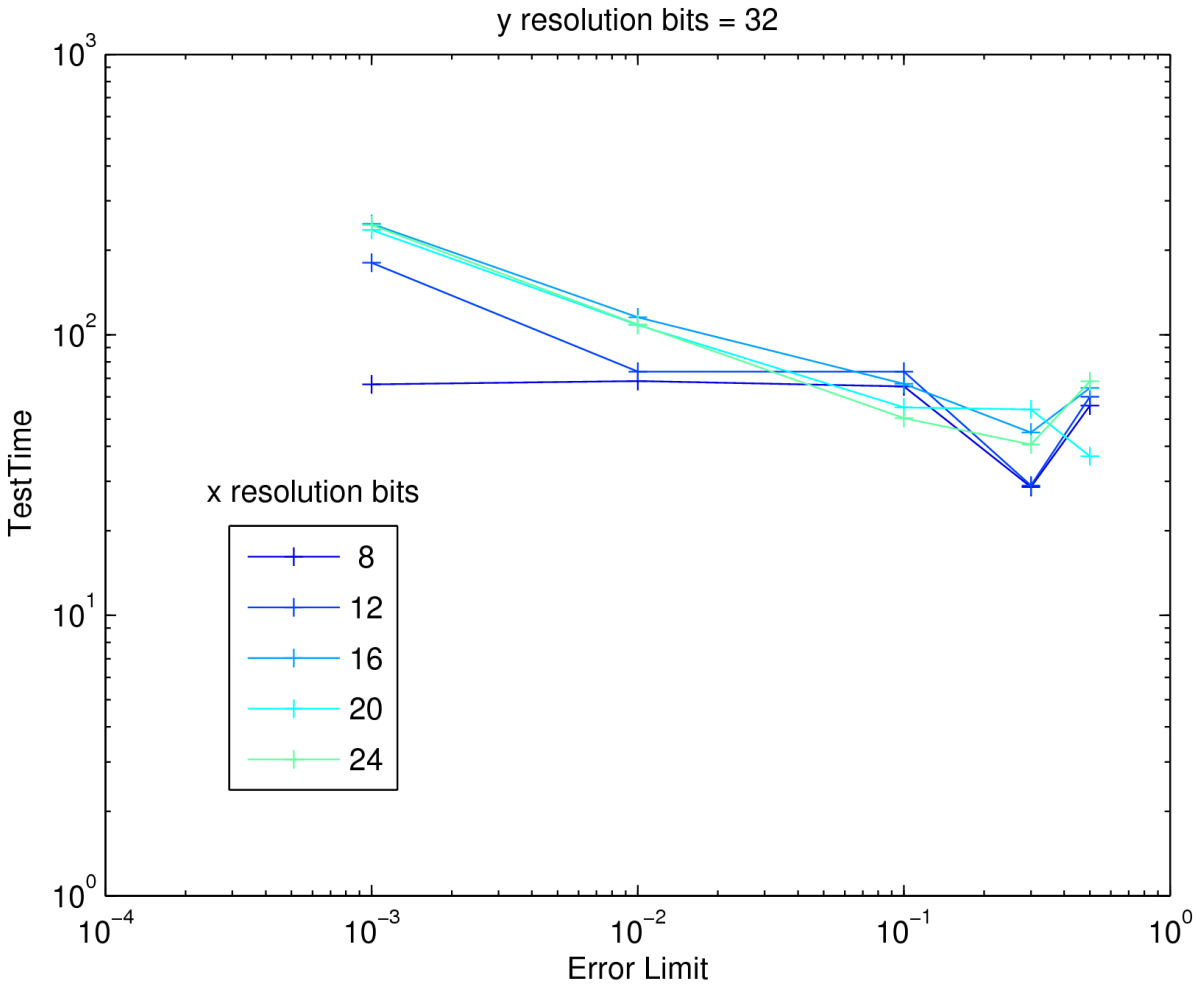
Costruzione del circuito (Java)



- Dipende in modo trascurabile dal numero di bit Q_y e Q_x utilizzati per rappresentare ingresso e uscita.
- Dipende molto dal livello di precisione ricercato ε
- Si notano leggere anomalie per risoluzioni molto basse

Conclusioni /4
Dati Sperimentali /4

Esecuzione del circuito (Java)



- Dipende in modo trascurabile dal numero di bit Q_y e Q_x utilizzati per rappresentare ingresso e uscita.
- Dipende dal livello di precisione ricercato ε (cioè dal numero di porte non-XOR)
- Si notano leggere anomalie per risoluzioni molto basse

Conclusioni /5

Lavoro Svolto:

-Software per la sintesi e test dei circuiti

- Java
- Paradigma funzionale
- formato per Fairplay (*.circuit *.fmt)
- formato personale (*.gc)

-Software per esecuzione Multiparty dei circuiti

- Java
- Implementazione FreeXOR e FreeNOT
- I/O interi di lunghezza arbitraria (BigInteger)

Sviluppi Futuri:

- Test su quattro funzioni di prova (arctg, gauss, sin)
- Dimensioni Circuito, numero spezzate, tempi sintesi ed esecuzioni al variare dell'errore e dei bit di rappresentazione