Extraction de Racine carrée matérielle

Alain GUYOT

Concurrent Integrated Systems
TIMA

☎ (33) 04 76 57 46 16

Alain.Guyot@imag.fr

http://tima-cmp.imag.fr/Homepages/guyot

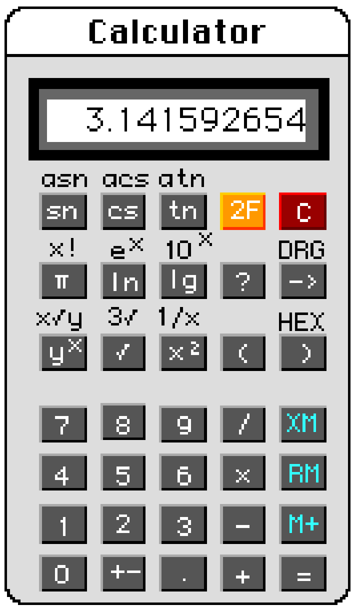
Techniques de l'Informatique et de la Microélectronique
pour l'Architecture. Unité associée au C.N.R.S. n° B0706

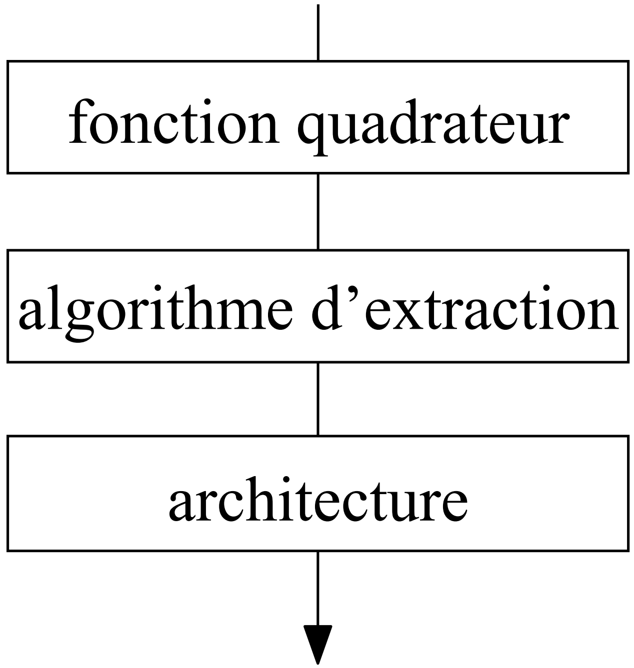
But: Réaliser des extracteurs de racine carrée combinatoires rapides

Optimiser la surface et/ou la vitesse

Problèmes
- Propagation de la retenue

Moyen
Utiliser des additionneurs sans propagation de retenue





Généralités sur la racine carrée

Nous calculerons la racine d'un nombre normalisé A tel que $1 \leq A < 4$. $A = 2 * a_{-1} + \sum_{i=0}^{n-1} a_i * 2^{-i}$

Alors $Q = \sqrt[2]{A}$ est également normalisé: $1 \leq Q < 2$.

De même que l'architecture d'un diviseur (naïf) se déduit de l'architecture d'un multiplieur (naïf) l'architecture d'un extracteur de racine carrée se déduit de celle d'un quadrateur.

Les algorithmes d'extraction de racine carrée se déduisent également des algorithmes de division car:

$$\text{si } Q = \frac{A}{Q} \text{ alors } Q = \sqrt[2]{A}$$

Algorithme naïf:

$Q_0 := 1$;
si $(Q_j + 2^{-j-1})^2 \leq A$ **alors** $Q_{j+1} := Q_j + 2^{-j-1}$
sinon $Q_{j+1} := Q_j$;

Iteration de Héron:

$$Q_{j+1} = \frac{1}{2}\left(Q_j + \frac{A}{Q_j}\right)$$

Quadrateur

On veut maintenir l'invariant $Q_{2j} = R_j^2 \quad \forall\ j \qquad R_j = \sum_{i=0}^{j} r_i 2^{-i}$

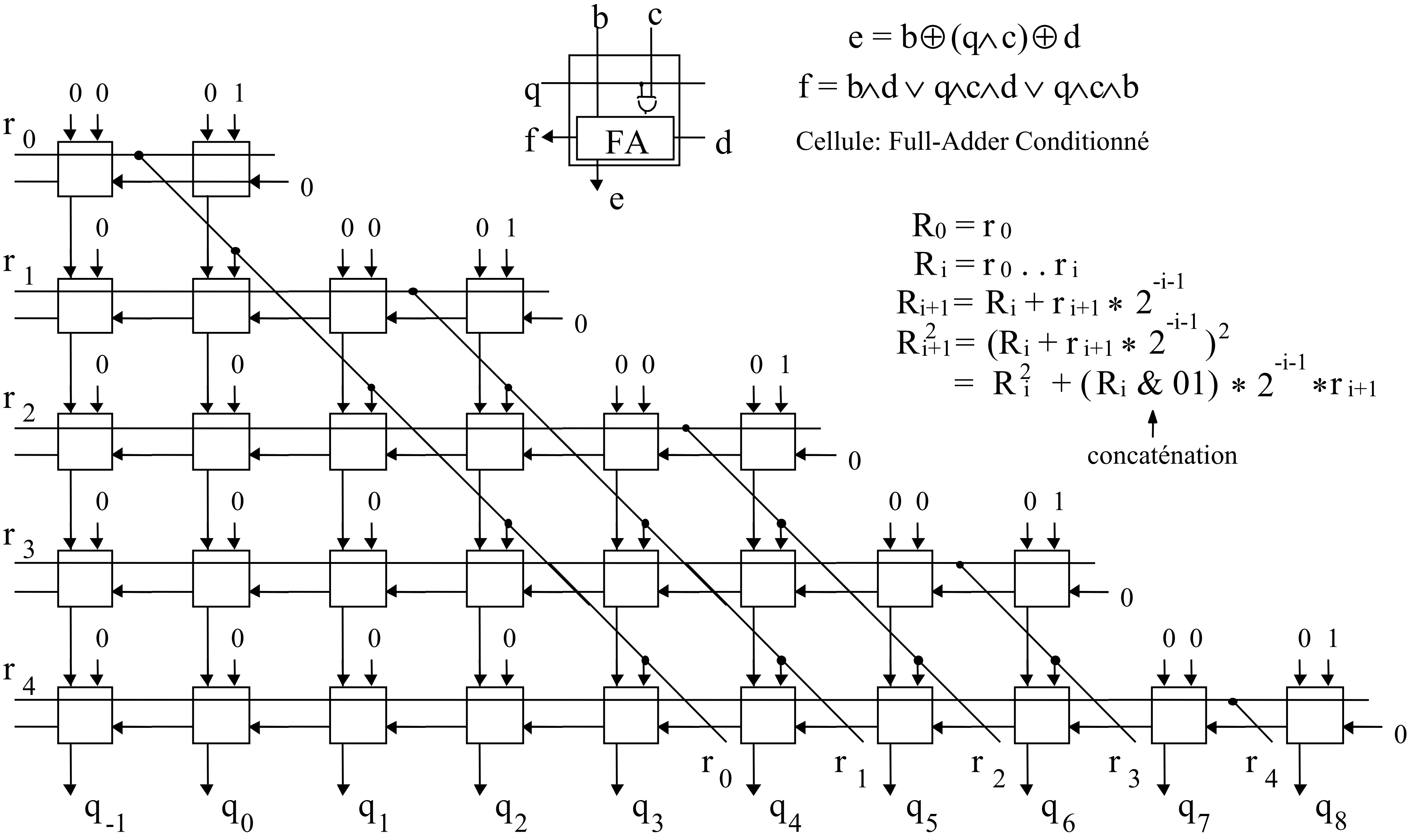
Donc quand $R_{j+1} := R_j + r_{j+1} * 2^{-j-1}$ alors

$$(R_{j+1})^2 := (R_j)^2 + r_{j+1} * (2 * R_j + 2^{-j-1}) * 2^{-j-1}$$

$R_0 := r_0$; $Q_0 := R_0$; $j := 0$;
Tantque $j \leq$ précision_requise **faire**
    $j := j+1$;
    $Q_{2j+2} := Q_{2j} + r_{j+1} * (2 * R_j + 2^{-j-1}) * 2^{-j-1}$;
    $R_{j+1} := R_j + r_{j+1} * 2^{-j-1}$
Fintantque

concaténations

Quadrateur à simplifier



$$e = b \oplus (q \wedge c) \oplus d$$

$$f = b \wedge d \vee q \wedge c \wedge d \vee q \wedge c \wedge b$$

Cellule: Full-Adder Conditionné

$$R_0 = r_0$$

$$R_i = r_0 \,.\,.\, r_i$$

$$R_{i+1} = R_i + r_{i+1} * 2^{-i-1}$$

$$R_{i+1}^2 = (R_i + r_{i+1} * 2^{-i-1})^2$$

$$= R_i^2 + (R_i \,\&\, 01) * 2^{-i-1} * r_{i+1}$$

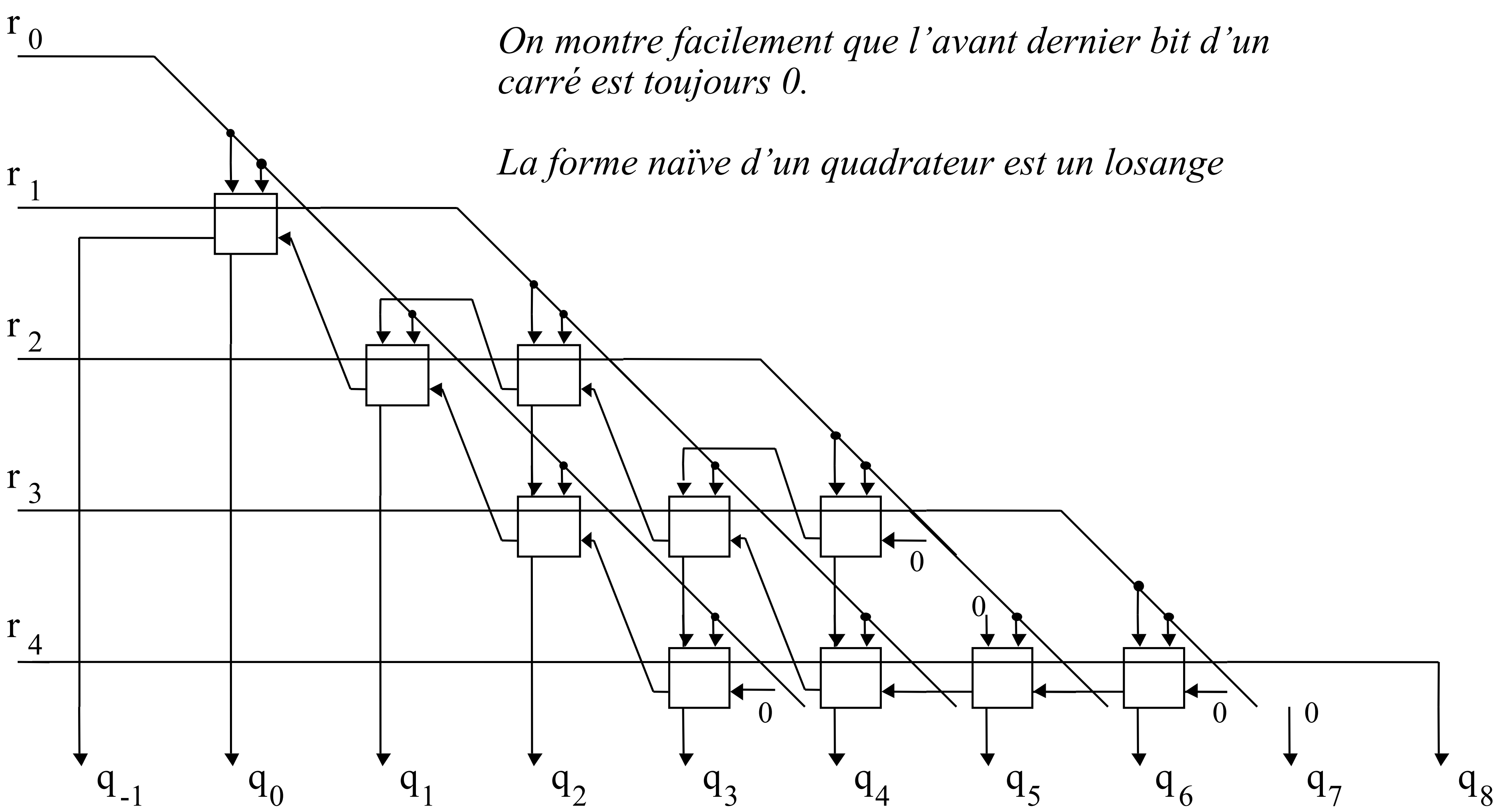
concaténation

Quadrateur

Elimination des FA dont la sortie est égale à l'entrée

On montre facilement que l'avant dernier bit d'un carré est toujours 0.

La forme naïve d'un quadrateur est un losange



Extracteur de racine carrée récurrent (sans restauration)

On va construire une suite $Q_0, Q_1, Q_2, \ldots Q_n$ et une suite $R_0, R_2, R_4, \ldots R_{2n}$

On veut maintenir l'invariant $(Q_j)^2 = A - R_{2j} \ \forall j \quad (R_{2j} \longrightarrow 0) \qquad \{ A \in [1,4[\}$

$R_{2j+2} := R_{2j} - Q_{j+1}^2 + Q_j^2 = R_{2j} - (Q_j + q_{j+1} * 2^{-j-1})^2 + Q_j^2$

$Q_1 := 1,1$; $R_2 := A - 1\,0,0\,1$; $j := 1$;

Tantque $j \leq$ précision_requise **faire**

  $j := j + 1$;

  si $R_{2j} \geq 0$ **alors** $\{ q_{j+1} = +1 \}$

    $Q_{j+1} := Q_j + 2^{-j-1}$; *concaténations*

    $R_{2j+2} := R_{2j} - (Q_j * 2^{-j} + 2^{-2j-2})$;

  sinon $\{ q_{j+1} = -1 \}$

    $Q_{j+1} := Q_j - 2^{-j-1}$;

    $R_{2j+2} := R_{2j} + (Q_j * 2^{-j} - 2^{-2j-2})$;

  finsinon ;

fintanque ;

Exemple d'extraction de racine carrée sans restauration

si $R_{2j} \geq 0$ **alors**

$R_{2j+2} := R_{2j} \; - Q_j * 2^{-j} - 2^{-2j-2}$

$Q_{j+1} := Q_j + 2^{j+1}$

sinon

$R_{2j+2} := R_{2j} \; + Q_j * 2^{-j} - 2^{-2j-2}$

$Q_{j+1} := Q_j - 2^{j+1}$

Calcul des restes		Valeurs du reste (signé)		Calcul des racines	Valeurs
R_0 := 1,111001 - 1,0		01,111001	1,890625	Q_0 := 1	1
R_2 := R_0 –1–0,01	:= R_0 – 1,01	00,111001	0,890625	Q_1 := Q_0 + 0,1	1,1
R_4 := R_2 +0,11-0,0001	:= R_2 + 0,1011	11,101001	- 0,359375	Q_2 := Q_1 – 0,01	1,01
R_6 := R_4 –0,0101-0,000001	:= R_4 – 0,010101	00,010101	0,328125	Q_3 := Q_2 + 0,001	1,011
R_8 := R_6 –		00,000000	0,000000	Q_4 := Q_3 + 0,0001	1,0111

Extracteur de racine carrée

$Q_1 := 1 * 2^{-1}$; $R_2 :=$ $(a_1 * 2^{-1}\ a_2 * 2^{-2})$ - $0{,}0\ 1$; $j := 1$;

Tantque $j \leq$ précision_requise **faire**

$j := j + 1$;

si $R_{2j} \geq 0$ **alors**

$q_{j+1} := +1$

sinon

$q_{j+1} := -1$

finsinon ;

$Q_{j+1} := Q_j + q_{j+1} * 2^{-j-1}$;

$R_{2j+2} := (R_{2j} + a_{2j+1} * 2^{-2j-1} + a_{2j+2} * 2^{-2j-2}) - q_{j+1} * (Q_j + q_{j+1} * 2^{-j-2}) * 2^{-j}$;

fintanque ;

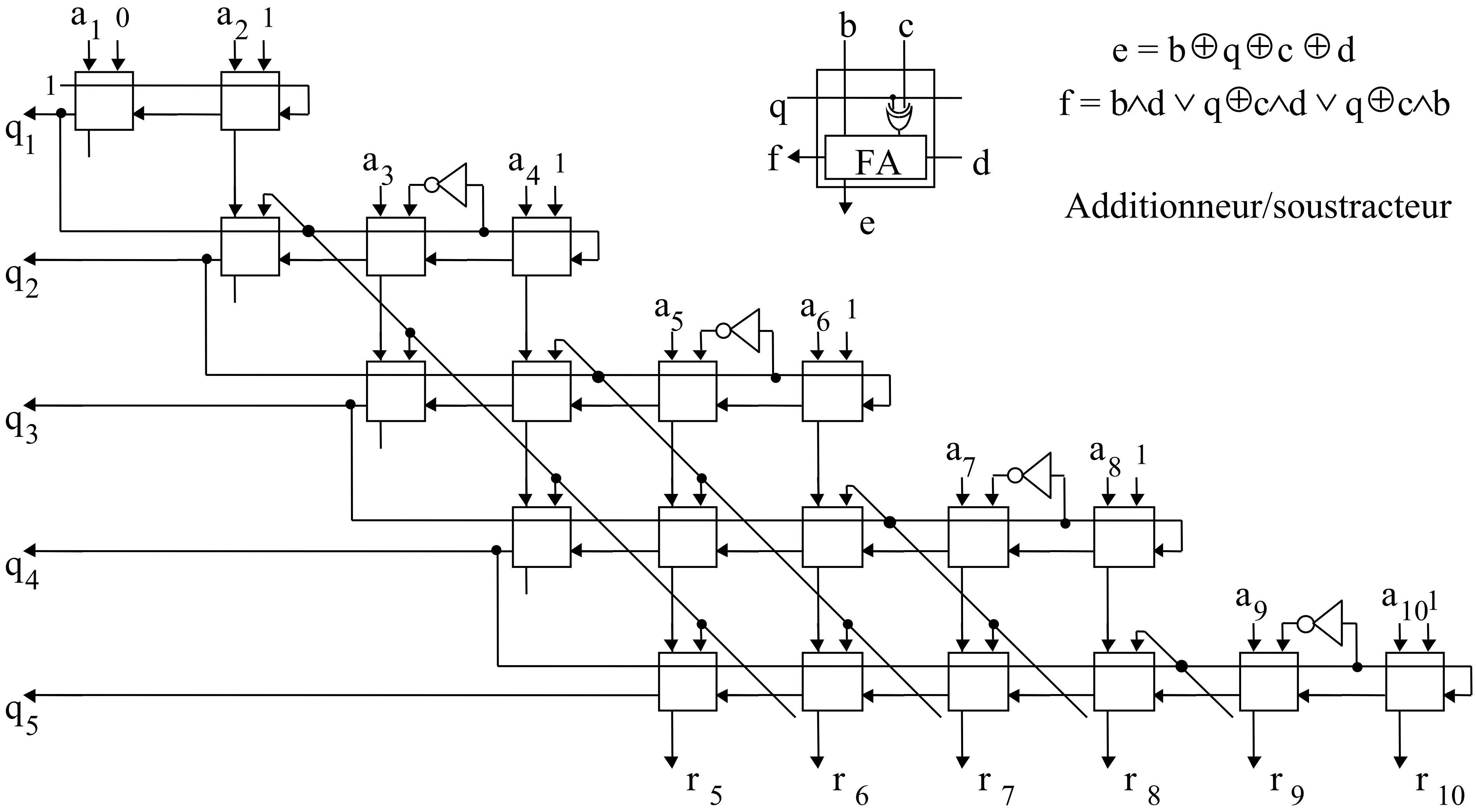
On code en binaire $p_{i+1} = 0$ si $q_{i+1} = +1$ et $p_{i+1} = 1$ si $q_{i+1} = -1$ (signe de q_{i+1}) .

Le cas général est p_{i+1}= retenue sortante de l'opération add/sous (signe)

$$R_{2j+2} := (R_{2j} + a_{2j+1} * 2^{-2j-1} + a_{2j+2} * 2^{-2j-2}) - (-1)^{p_{j+1}} (Q_{j-1}\ \&\ p_{j+1}\ \overline{p_{j+1}}\ 1) * 2^{-j}$$

Extracteur de racine carrée (2)

sans restauration



$e = b \oplus q \oplus c \oplus d$

$f = b \wedge d \vee q \oplus c \wedge d \vee q \oplus c \wedge b$

Additionneur/soustracteur

Extracteur de racine carrée(3)

(examen des 3 premiers chiffres significatifs de R^{2j})

$Q_0 := 1 * 2^{-0}$; $R_0 := (a_{-1} * 2^1 \ a_{-2} * 2^2) - 1$; $j := 0$;

Tantque $j \leq$ précision_requise **faire**

$j := j + 1$;

si le signe de R_{2j} n'est pas distinguable par l'examen de ses 3 premier chiffres **alors**

$q_{j+1} := 0$

sinon si $R_{2j} \geq 0$ **alors**

$q_{j+1} := +1$

sinon

$q_{j+1} := -1$

finsinon ;

$Q_{j+1} := Q_j + q_{j+1} * 2^{-j-1}$;

$R_{2j+2} := (R_{2j} + a_{2j+1} * 2^{-2j-1} + a_{2j+2} * 2^{-2j-2}) - q_{j+1} * (Q_j - q_{j+1} * 2^{-j-2}) * 2^{-j}$;

fintanque ;

Calcul du reste partiel R_{2j+2}

Itération générale: $R_{2j+2} := (R_{2j} + a_{2j+1} * 2^{-2j-1} + a_{2j+2} * 2^{-2j-2}) - q_{j+1} * (Q_j - q_{j+1} * 2^{-j-2}) * 2^{-j}$;

Si $q_{j+1} = 0$ alors $R_{2j+2} := (R_{2j} + a_{2j+1} * 2^{-2j-1} + a_{2j+2} * 2^{-2j-2})$;

Si $q_{j+1} = +1$ alors $R_{2j+2} := (R_{2j} + a_{2j+1} * 2^{-2j-1} + a_{2j+2} * 2^{-2j-2}) - Q_j * 2^{-j} + 2^{-2j-2}$; {soustraction}

Si $q_{j+1} = -1$ alors $R_{2j+2} := (R_{2j} + a_{2j+1} * 2^{-2j-1} + a_{2j+2} * 2^{-2j-2}) + Q_j * 2^{-j} + 2^{-2j-2}$; {addition}

Par concaténation on a: $Q_j = \sum_{i=-1}^{j} q_i * 2^{-i}$ avec $q_i \in \{-1, 0, +1\}$

Pour effectuer l'addition/soustraction, on désire $Q_j = \sum_{j=-1}^{j} p_i * 2^{-i}$ avec $p_i \in \{0, +1\}$

Solution: convertir Q_j “à la volée” (changer la représentation en conservant la valeur).

Conversion de la racine carrée

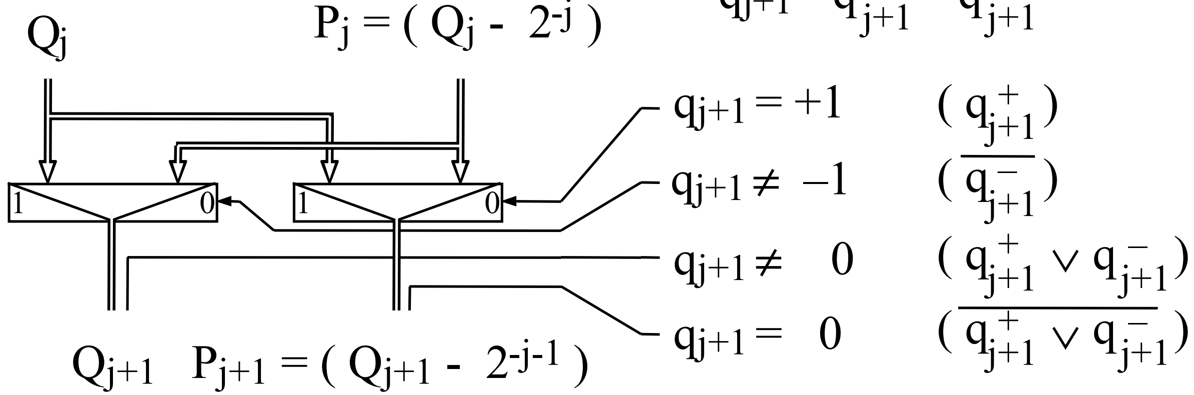
Par concaténation on a: $Q_j = \sum_{i=-1}^{j} q_i * 2^{-i}$ avec $q_i \in \{ -1, 0, +1\}$

Pour effectuer l'addition/soustraction, on désire $Q_j = \sum_{i=-1}^{j} p_i * 2^{-i}$ avec $p_i \in \{0, +1\}$

Solution: calculer simultanément les valeurs Q_i et $P_j = (Q_j - 2^{-j}) \; \forall \; j$.
On a Q_j et $(Q_j - 2^{-j})$ et on veut obtenir Q_{j+1} et $(Q_{j+1} - 2^{-j-1})$ sans propagation de retenue

Si $q_{j+1} = -1$ alors $Q_{j+1} := (Q_j - 2^{-j}) + 2^{-j-1}$; $(Q_{j+1} - 2^{-j-1}) := (Q_j - 2^{-j})$;
Si $q_{j+1} = 0$ alors $Q_{j+1} := Q_j$; $(Q_{j+1} - 2^{-j-1}) := (Q_j - 2^{-j}) + 2^{-j-1}$;
Si $q_{j+1} = +1$ alors $Q_{j+1} := Q_j + 2^{-j-1}$; $(Q_{j+1} - 2^{-j-1}) := Q_j$;

$q_{j+1} = q^{+}_{j+1} - q^{-}_{j+1}$



Extraction de racine carrée sans propagation

$Q_1 := 1 * 2^{-1}$; $P_1 := O$; $R_2 := (a_1 * 2^{-1}\ a_2 * 2^{-2}) - 0{,}0\ 1$; $j := 1$;

Tantque $j \leq$ précision_requise **faire**

$j := j + 1$;

si le signe de R_{2j} n'est pas distinguable par l'examen de ses 3 premier chiffres **alors**

$Q_{j+1} := Q_j$;

$P_{j+1} := P_j + 2^{-j-1}$;

$R_{2j+2} := (R_{2j} + a_{2j+1} * 2^{-2j-1} + a_{2j+2} * 2^{-2j-2})$;

sinon si $R_{2j} > 0$ **alors**

$Q_{j+1} := Q_j + 2^{-j-1}$;

$P_{j+1} := Q_j$;

$R_{2j+2} := (R_{2j} + a_{2j+1} * 2^{-2j-1} + a_{2j+2} * 2^{-2j-2}) - Q_j * 2^{-j} + 2^{-2j-2}$;

sinon

$Q_{j+1} := P_j + 2^{-j-1}$;

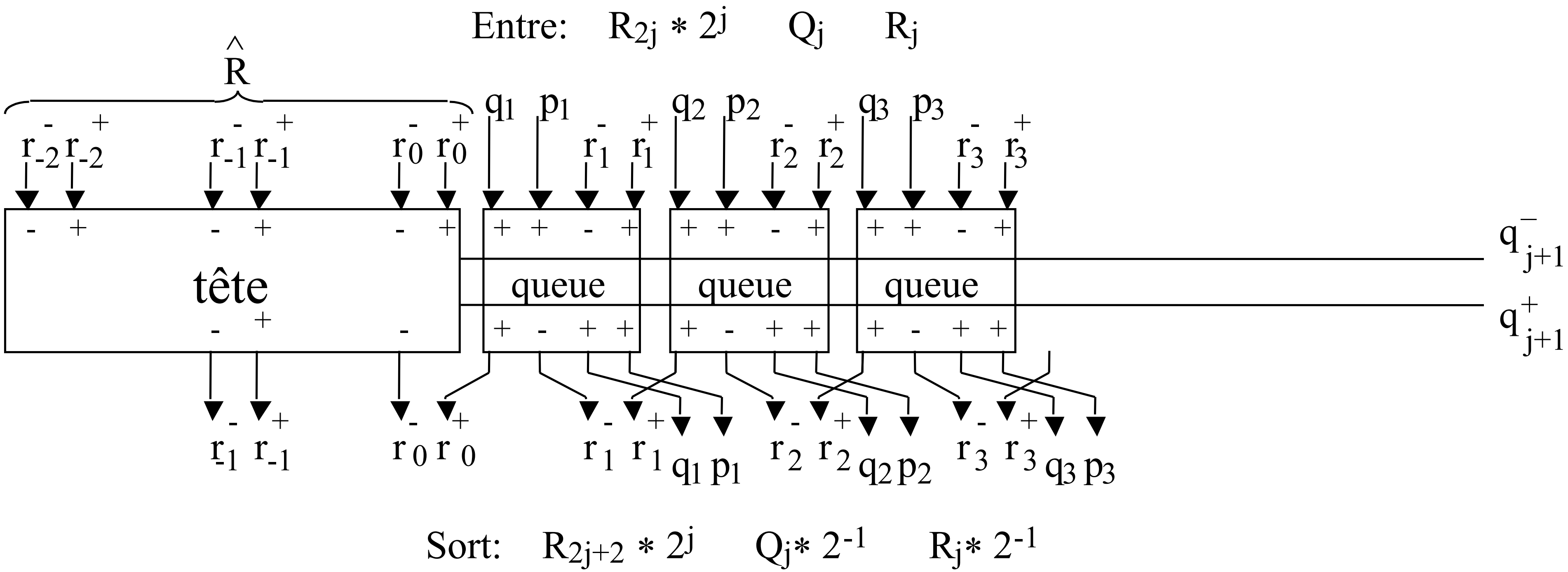
$P_{j+1} := P_j$;

$R_{2j+2} := (R_{2j} + a_{2j+1} * 2^{-2j-1} + a_{2j+2} * 2^{-2j-2}) + Q_j * 2^{-j} + 2^{-2j-2}$;

finsinon ;

fintanque ;

Rôles de la tête et de la queue



Tête:

1- Déterminer l'opération à exécuter
q_{j+1} = (add, sous ou rien)

2- Exécuter cette opération sur les chiffres de tête

3- Recoder le résultat pour éliminer le chiffre poids fort r_{-2} en l'annulant

Queue:

1- Exécuter l'opération q_{j+1} = (add, sous ou rien) sur les chiffres de queue sans retenue propagé

2- Transcoder Q_{j+1} de notation redondante BS à conventionnelle

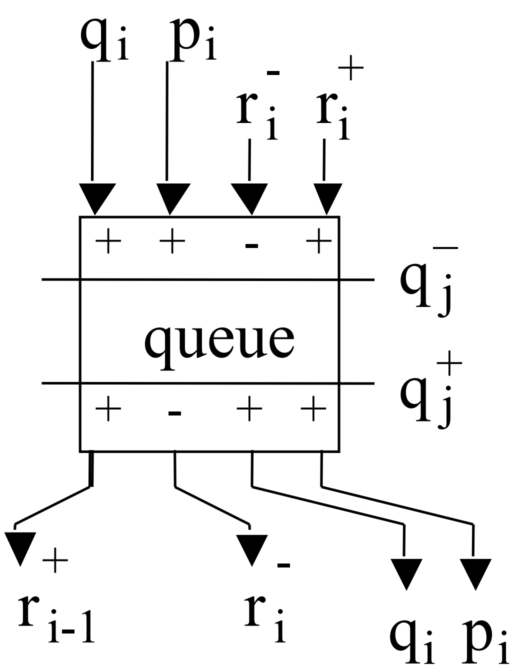
Équations de la queue

Remarques:

1- Pour chaque tranche, R_{2j+2} a 1 chiffre de moins à gauche et 2 chiffres de plus à droite que R_{2j}.

2- Pour chaque tranche Q_{j+1} et P_{j+1} ont un bit de plus à droite que Q_j et P_j.

3- $P_j = (Q_j - 2^{-j})$. Donc $-Q_j = -P_j - 2^{-j} = \overline{P_j} + 2^{-j} - 2^{-j} = \overline{P_j}$.



Equation arithmétique

$$2 * r_{i-1}^+ + \overline{r_i^-} = r_i^+ + \overline{r_i^-} + (q_j^+ \wedge \overline{p_i} \vee q_j^- \wedge q_i)$$

Conversion

$$q_i = (q_j^- \wedge p_i \vee \overline{q_j^-} \wedge q_i)$$

$$p_i = (q_j^+ \wedge p_i \vee \overline{q_j^+} \wedge q_i)$$

Conclusion

Un algorithme d’extraction de racine carrée rapide a été proposé:

- Les Additions/Soustraction ne propagent pas la retenue

- Le calcul, le recodage et le test du reste partiel sont effectués simultanément (et non séquentiellement)

- Les lignes longues et leurs amplis sont éliminées du chemin critique

- L’algorithme de conversion "a la volée" de la racine carrée, introduit dans la division (page ..), peut être utilisé.