Automated Search for Round 1 Differentials for SHA-1

Phil Hawkes, Michael Paddon, Greg Rose

QUALCOMM

{phawkes,mwp,ggr}@qualcomm.com

Motivation for Research

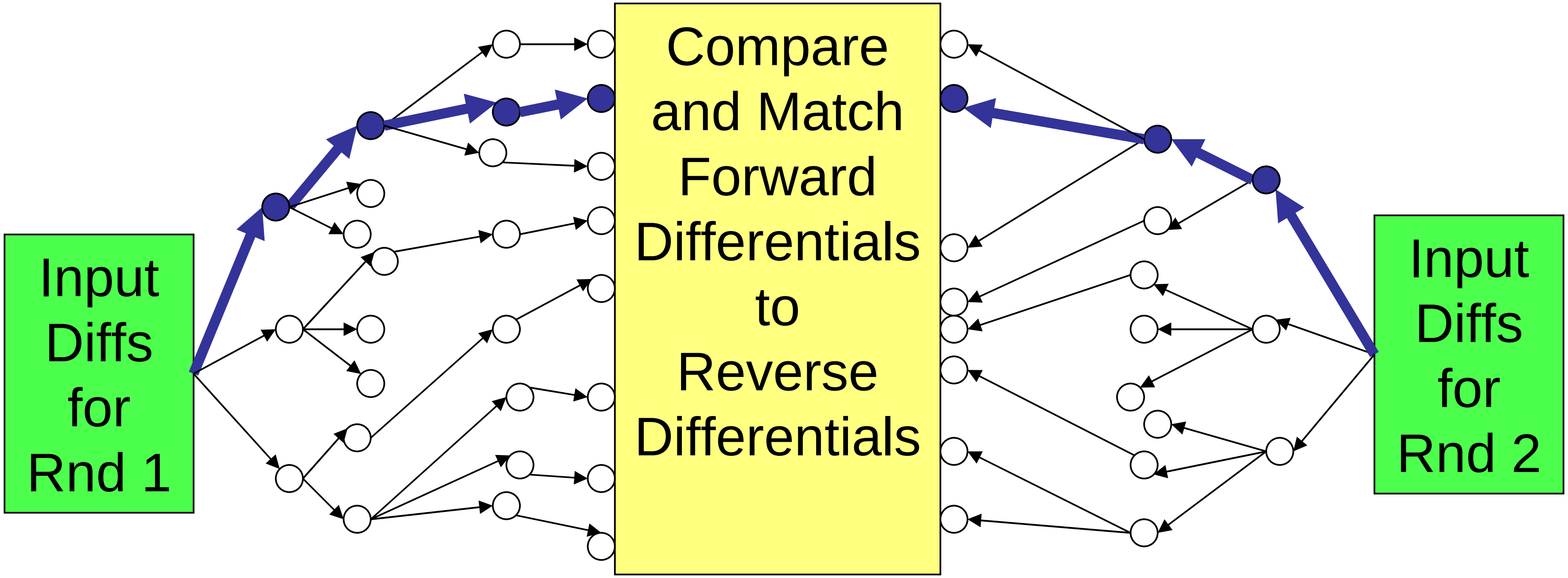
- Given:
  - Disturbance vector (XOR diffs in msg words),
  - Input difference to Round 1,
  - Input difference for Round 2, …
- …is there a differential path?
- Which Round 1differential path is optimal?
  - E.g. improvements to MD5 attacks
- How do we find optimal paths?
  - **Automate search**!

1 2 3 4 ● ● ● s s+1 ● ● ● 18 19 20

Generate set of
FORWARD Differentials
Steps 1 to s

Generate set of
REVERSE Differentials
Steps 20 to (s+1)

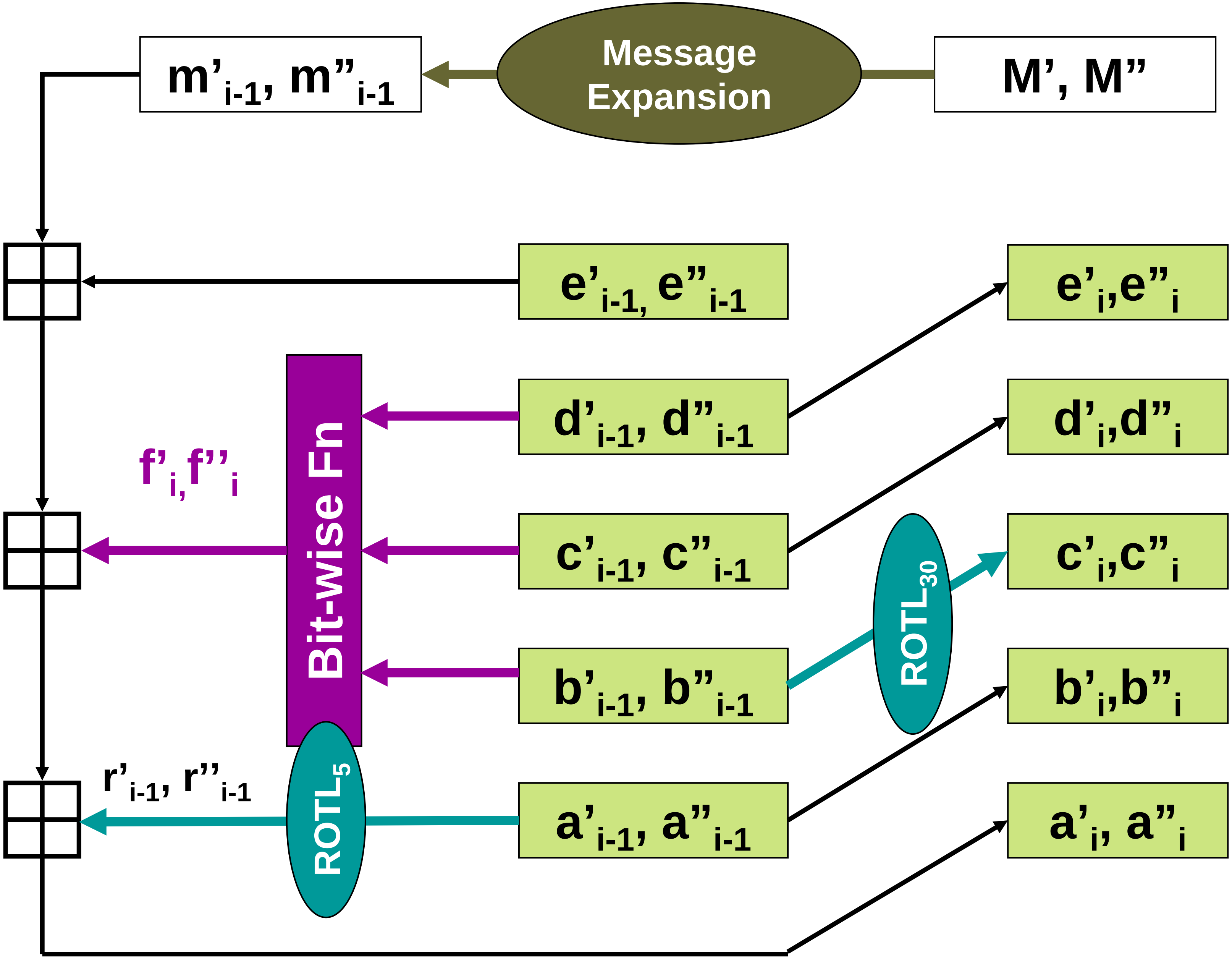
Sequence of XOR Diffs $\Delta_{\oplus}$ m for Steps 1-20



Input
Diffs
for
Rnd 1

Compare
and Match
Forward
Differentials
to
Reverse
Differentials

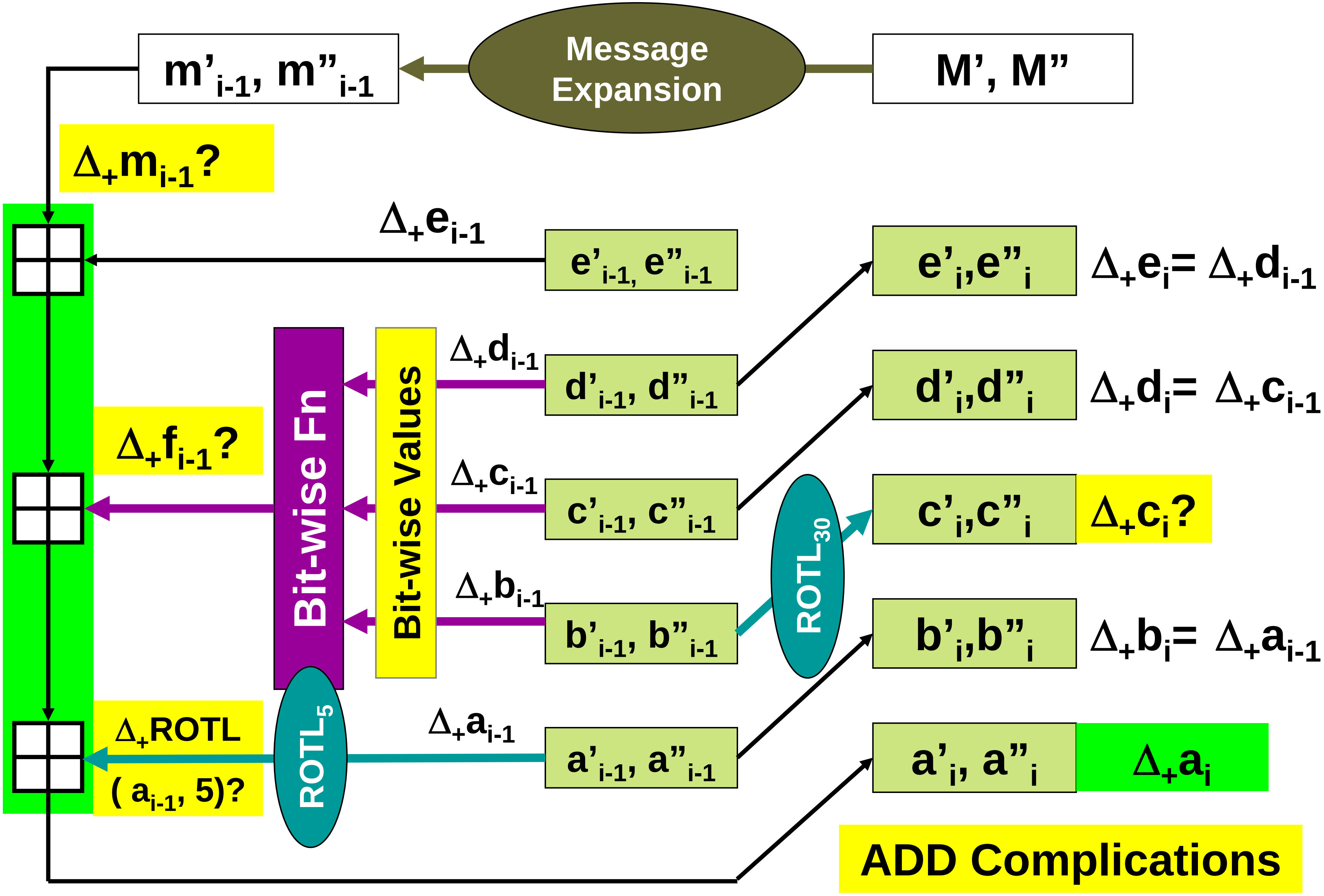
Input
Diffs
for
Rnd 2



Message Expansion
M', M"
m'$_{i-1}$, m"$_{i-1}$
e'$_{i-1,}$ e"$_{i-1}$
d'$_{i-1}$, d"$_{i-1}$
c'$_{i-1}$, c"$_{i-1}$
b'$_{i-1}$, b"$_{i-1}$
a'$_{i-1}$, a"$_{i-1}$
e'$_{i}$,e"$_{i}$
d'$_{i}$,d"$_{i}$
c'$_{i}$,c"$_{i}$
b'$_{i}$,b"$_{i}$
a'$_{i}$, a"$_{i}$
Bit-wise Fn
f'$_{i,}$f''$_{i}$
ROTL$_{30}$
ROTL$_{5}$
r'$_{i-1}$, r''$_{i-1}$

ADD & XOR Differences

- **ADD difference**
  - $\mathbf{\Delta_{+}X = X'' - X'}$ (mod 2^{32})
- **XOR Difference**
  - $\mathbf{\Delta_{\oplus}X = X'' \oplus X'}$
- **Properties**
  - $\mathbf{\Delta_{+}(X+Y) \quad = \Delta_{+}X \quad + \quad \Delta_{+}Y}$
  - $\mathbf{\Delta_{\oplus}(X \oplus Y) \quad = \Delta_{\oplus}X \quad \oplus \quad \Delta_{\oplus}Y}$
  - $\mathbf{\Delta_{\oplus}ROTL(\,X,r\,) \quad = ROTL(\,\Delta_{\oplus}X,\,r\,)}$**:** r *fixed*



M’, M”

Message Expansion

m'_{i-1}, m''_{i-1}

$\Delta_+ m_{i-1}$?

$\Delta_+ e_{i-1}$

e'_{i-1}, e''_{i-1}

$\Delta_+ d_{i-1}$

d'_{i-1}, d''_{i-1}

$\Delta_+ c_{i-1}$

c'_{i-1}, c''_{i-1}

$\Delta_+ b_{i-1}$

b'_{i-1}, b''_{i-1}

$\Delta_+ a_{i-1}$

a'_{i-1}, a''_{i-1}

Bit-wise Fn

Bit-wise Values

$\Delta_+ f_{i-1}$?

ROTL_5

Δ_+ROTL (a_{i-1}, 5)?

ROTL_{30}

e'_i, e''_i $\Delta_+ e_i = \Delta_+ d_{i-1}$

d'_i, d''_i $\Delta_+ d_i = \Delta_+ c_{i-1}$

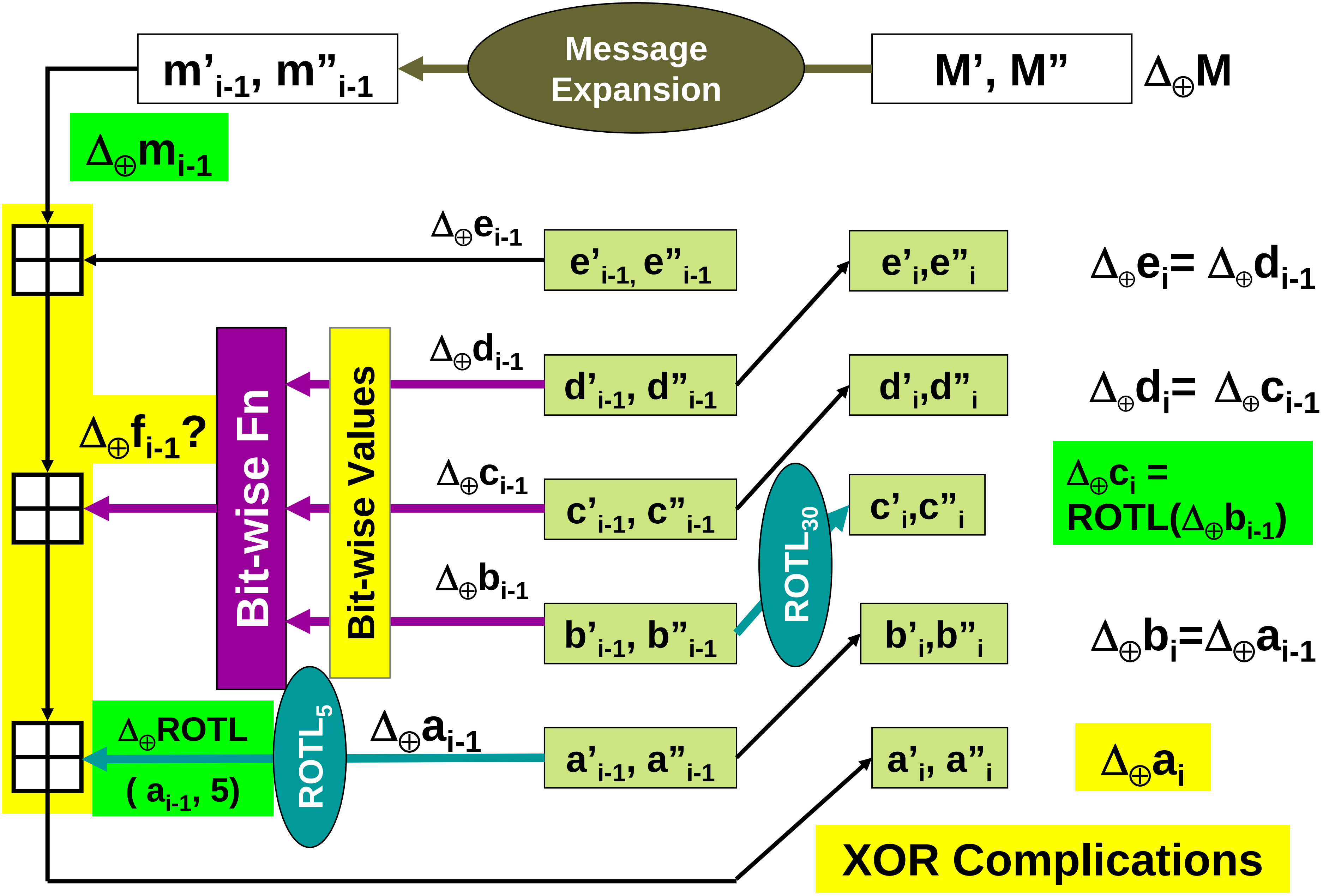
c'_i, c''_i $\Delta_+ c_i$?

b'_i, b''_i $\Delta_+ b_i = \Delta_+ a_{i-1}$

a'_i, a''_i $\Delta_+ a_i$

ADD Complications

XOR Complications



Nabla representation ∇X

- ∇X[j] =
  - @ if **X”[j]** ≠ **X’[j]**
  - **+** if (**X”[j],X’[j]**) = (1,0) ↔ **X”[j]- X’[j]** = +1
  - **-** if (**X”[j],X’[j]**) = (0,1) ↔ **X”[j]- X’[j]** = -1
  - ***** if **X”[j]** = **X’[j]**
  - **0** if **X”[j]** = **X’[j]** = 0
  - **1** if **X”[j]** ≠ **X’[j]** = 1
- $\boldsymbol{\Delta_{+}X} = \sum_{\mathbf{+,-}} \nabla X[j]\, 2^{j}$

Example

```
Bit3322222222221111111111
   10987654321098765432109876543210

X′=00111010101010100101101010101000

X″=10101010011010010101100100101000

∇X=+01-1010-+1010-+010110-+-0101000

Δ⊕=10010000110000110000001110000000
```

$\Delta + = \ +2^{31} \ -2^{28} \ -2^{23} \ +2^{22} \ -2^{17} \ +2^{16} \ -2^{9} \ +2^{8} \ -2^{7}$

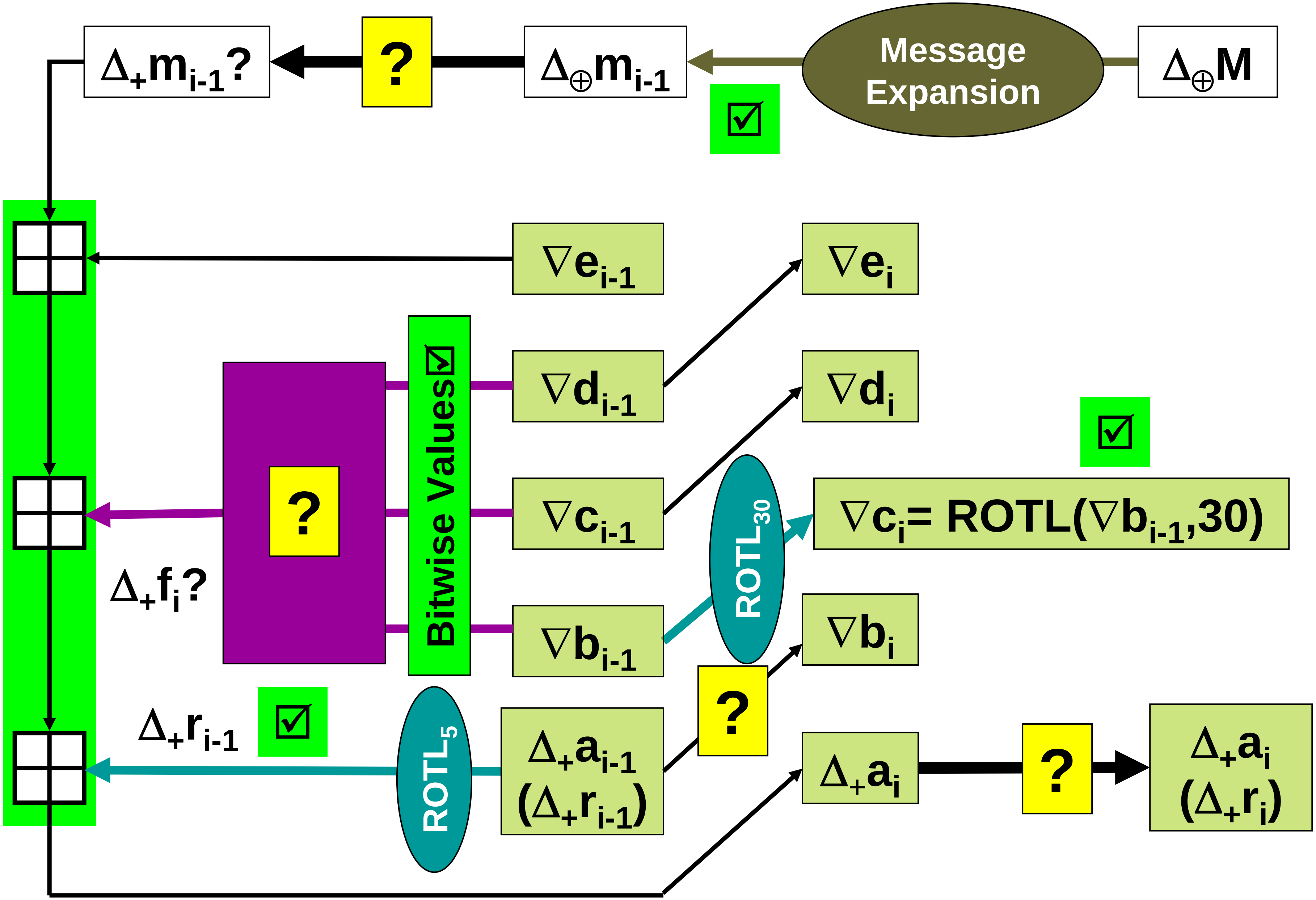
= 1874787968 = 0x6FBEFE80

= 01101111 10111110 11111110 10000000

- $X', X'' \rightarrow \nabla X \rightarrow \Delta_{+}X, \Delta_{\oplus}X$

Observations

- $\nabla \, \mathrm{ROTL}(X,r) = \mathrm{ROTL}(\nabla X,r)$
- XOR diffs only where **@,+,-**
  - **@,+,-** = **dynamic** bits
  - ***,1,0** = **static** bits
- ADD diff fully defined by **+,-** (& **@** MSB only)
- Values of static bits don’t affect XOR diff or ADD diff
  - Static bits only of interest in IF function


$\Delta_+ m_{i-1}$?
?
$\Delta_\oplus m_{i-1}$
☑
Message Expansion
$\Delta_\oplus M$
∇e_{i-1}
∇e_i
Bitwise Values☑
∇d_{i-1}
∇d_i
☑
?
∇c_{i-1}
ROTL_{30}
$\nabla c_i = \mathrm{ROTL}(\nabla b_{i-1},30)$
$\Delta_+ f_i$?
∇b_{i-1}
∇b_i
$\Delta_+ r_{i-1}$
☑
ROTL_5
$\Delta_+ a_{i-1}$ $(\Delta_+ r_{i-1})$
?
$\Delta_+ a_i$
?
$\Delta_+ a_i$ $(\Delta_+ r_i)$

Branching Points

- Given **XOR** diff, ∃ multiple **ADD** diffs
- Given **ADD** diff, ∃ multiple **ADD** diffs for **ROTL**
- Given **ADD** diff, ∃ multiple **XOR** diffs
- Given **XOR & ADD diff** in, ∃ multiple **ADD** diff out (IF)

Know:	Want:	Fn	Choice:
$\Delta_{\oplus}m$	$\Delta_{+}m$		**?**
$\Delta_{+}a$	$\Delta_{+}r$	**ROTL**	**?**
$\Delta_{+}a$, $\Delta_{+}r$	∇b		**?**
∇b, ∇c, ∇d	$\Delta_{+}f$	**IF**	**?**

Given **XOR** diffs find **ADD** diffs

Bit332222222222…
109876543210…

Δ⊕=0001000011 00…0

∇0=*+****++**…** **Δ+=**+2^{28} +2^{23} +2^{22}

∇1=*+****+-**…** **Δ+=**+2^{28} +2^{23} -2^{22}

∇2=*+****-+**…** **Δ+=**+2^{28} -2^{23} +2^{22}

∇3=*+****--**…** **Δ+=**+2^{28} -2^{23} -2^{22}

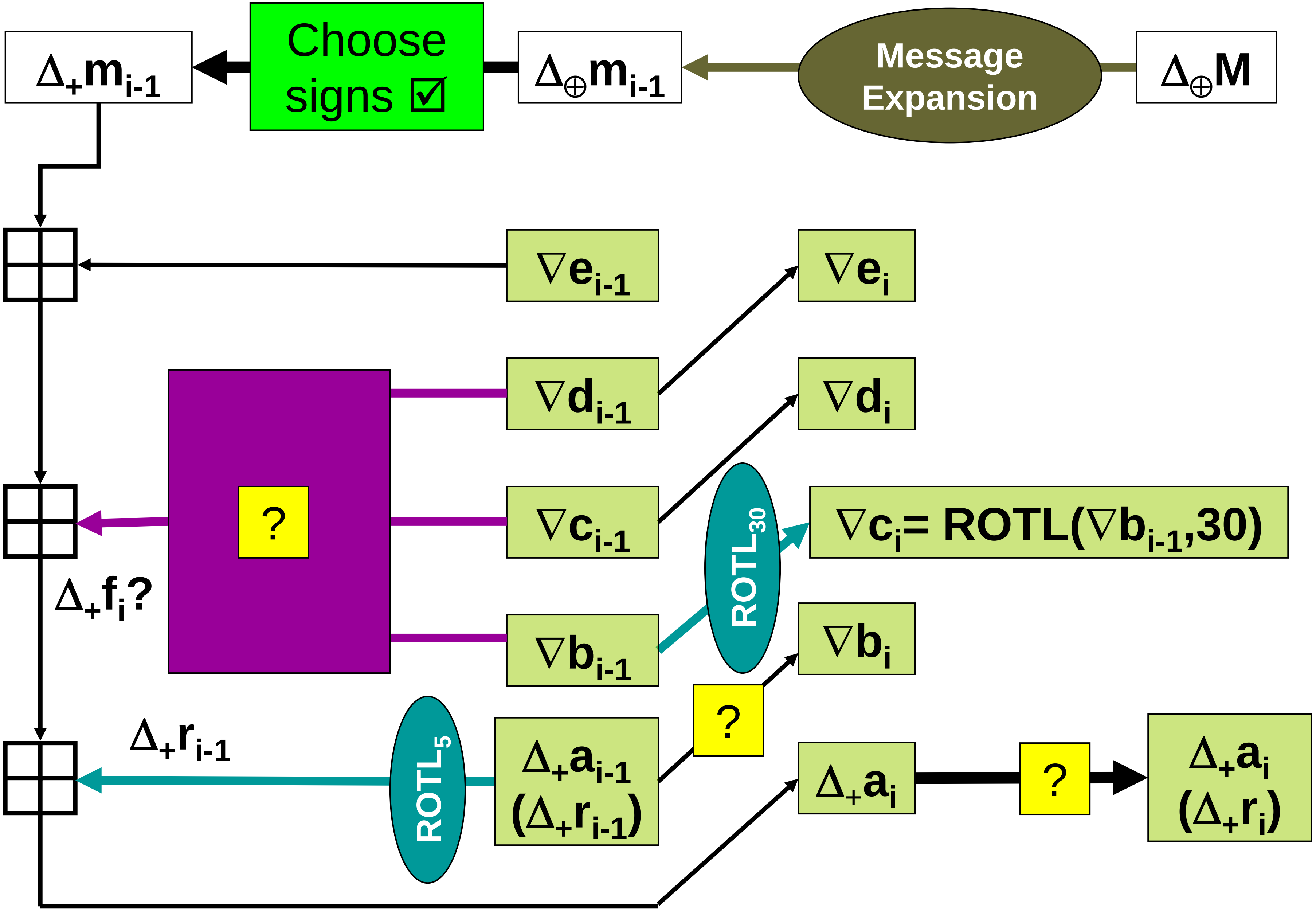
∇4=*-****++**…** **Δ+=**-2^{28} +2^{23} +2^{22}

∇5=*-****+-**…** **Δ+=**-2^{28} +2^{23} -2^{22}

∇6=*-****-+**…** **Δ+=**-2^{28} -2^{23} +2^{22}

∇7=*-****--**…** **Δ+=**-2^{28} -2^{23} -2^{22}

Each is a distinct addition difference



$\Delta_{+}m_{i-1}$

Choose signs ☑

$\Delta_{\oplus}m_{i-1}$

Message Expansion

$\Delta_{\oplus}M$

∇e_{i-1}

∇e_{i}

∇d_{i-1}

∇d_{i}

?

∇c_{i-1}

ROTL_{30}

$\nabla c_{i} = \mathrm{ROTL}(\nabla b_{i-1},30)$

$\Delta_{+}f_{i}$?

∇b_{i-1}

∇b_{i}

?

$\Delta_{+}r_{i-1}$

ROTL_{5}

$\Delta_{+}a_{i-1}$ $(\Delta_{+}r_{i-1})$

$\Delta_{+}a_{i}$

?

$\Delta_{+}a_{i}$ $(\Delta_{+}r_{i})$

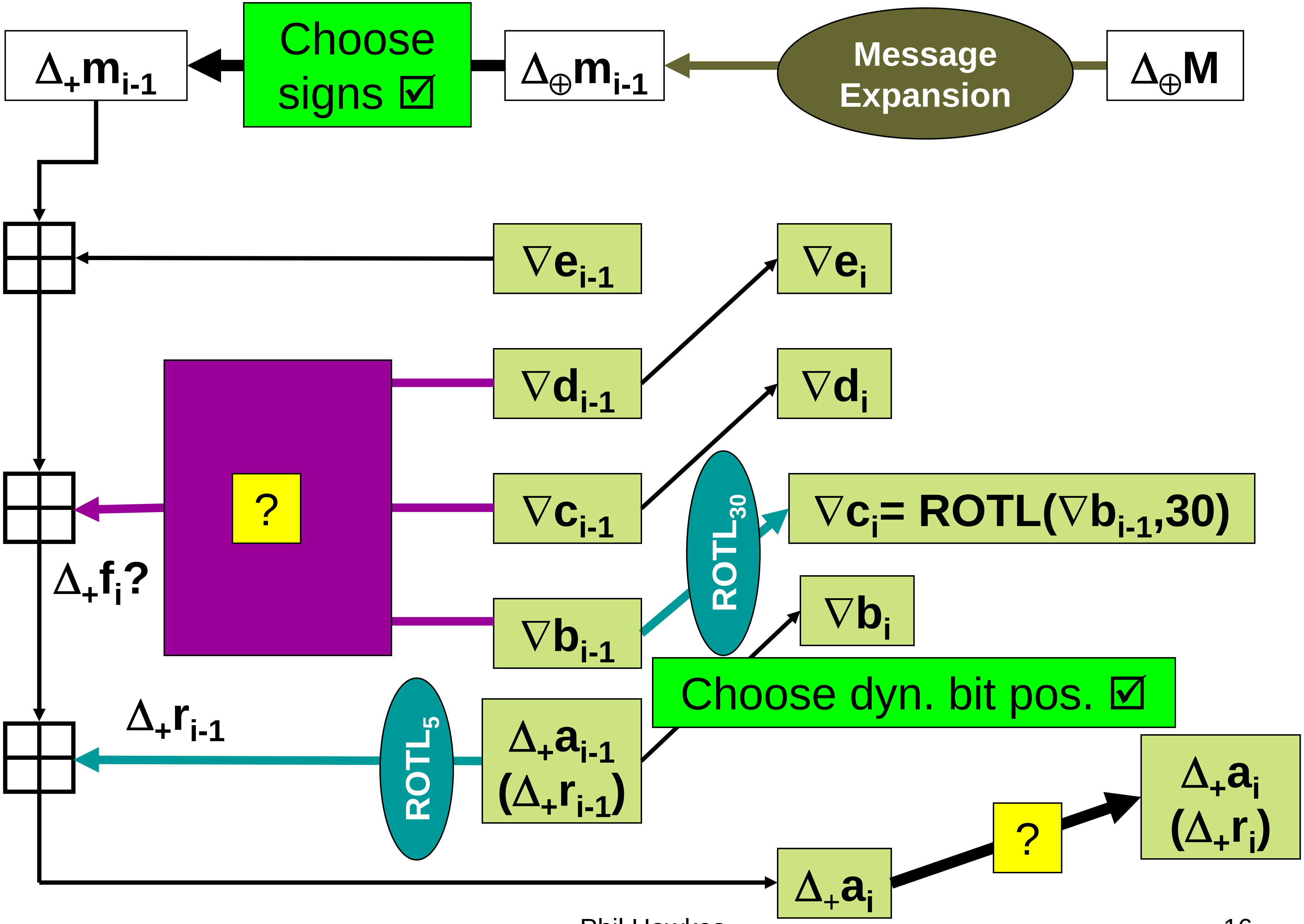
Given Δ+=2^{28} + 2^{25} find XOR diffs

∇0=***+**+*…	Δ⊕=00010010…
∇1=***+*+-*…	Δ⊕=00010110…
∇2=***++--*…	Δ⊕=00011110…
∇3=**+*---*…	Δ⊕=00101110…
∇4=**+-**+*…	Δ⊕=00110010…
∇5=**+-*+-*…	Δ⊕=00110110…
∇6=**+-+--*…	Δ⊕=00111110…
∇7=*+--**+*…	Δ⊕=01110010…
∇8=*+--*+-*…	Δ⊕=01110110…
∇9=*+--+--*…	Δ⊕=01111110…
∇A=*+-*---*…	Δ⊕=01101110…
∇B=+---**+*…	Δ⊕=11110010…
∇C=----**+*…	Δ⊕=11110010…
∇D=+---*+-*…	Δ⊕=11110110…
∇E=----*+-*…	Δ⊕=11110110…

- Carry addition differences up to higher order bits
- Cancel with existing higher order differences

  or…
- Add to higher order differences

etc



$\Delta_+ m_{i-1}$

Choose signs ☑

$\Delta_\oplus m_{i-1}$

Message Expansion

$\Delta_\oplus M$

∇e_{i-1}

∇e_i

∇d_{i-1}

∇d_i

?

∇c_{i-1}

ROTL_{30}

∇c_i= ROTL(∇b_{i-1},30)

$\Delta_+ f_i$?

∇b_{i-1}

∇b_i

Choose dyn. bit pos. ☑

$\Delta_+ r_{i-1}$

ROTL_5

$\Delta_+ a_{i-1}$ ($\Delta_+ r_{i-1}$)

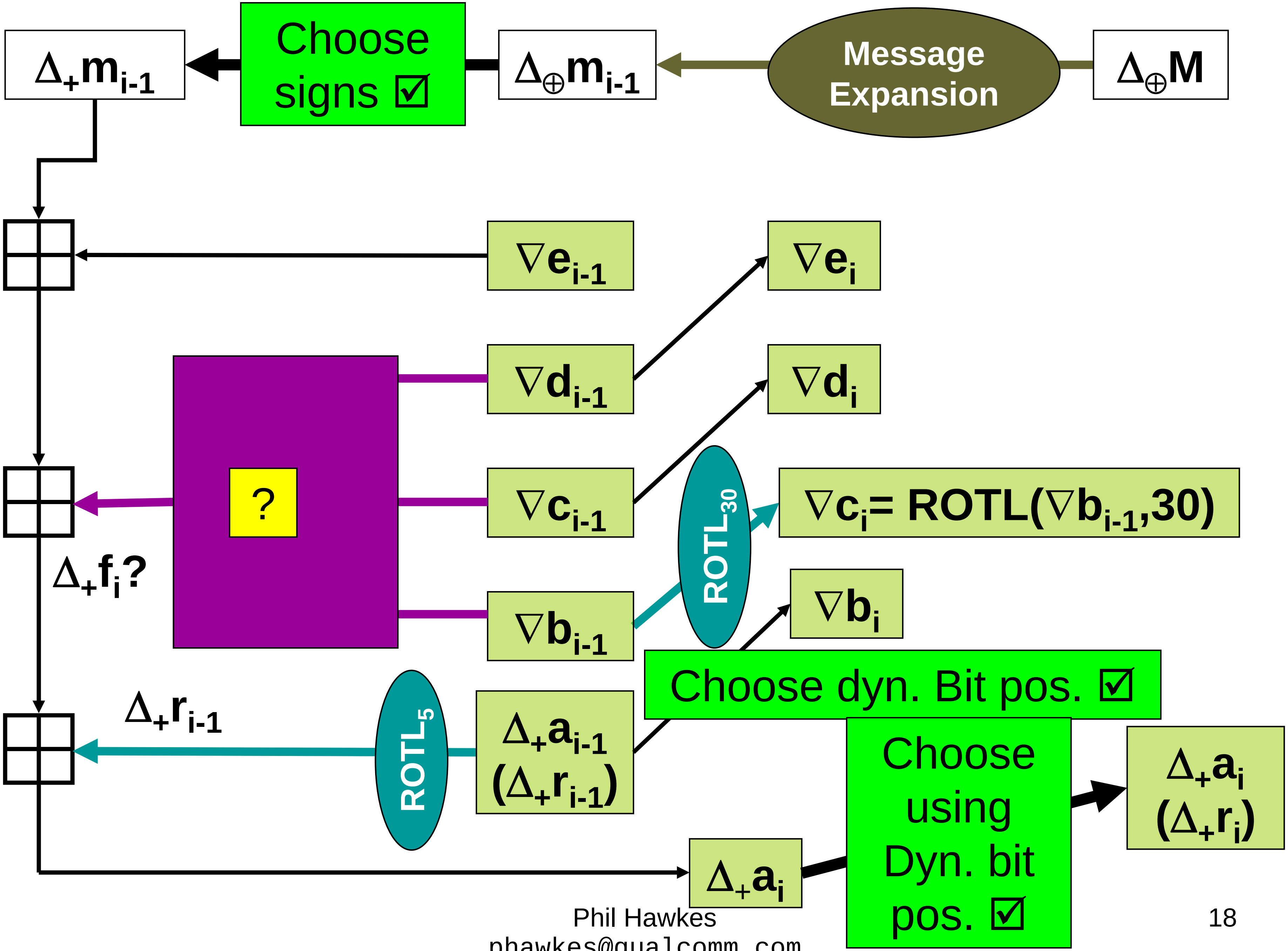
?

$\Delta_+ a_i$ ($\Delta_+ r_i$)

$\Delta_+ a_i$

$\Delta_+ a_{i-1} = 2^{28} + 2^{25}$, $\Delta_+\mathrm{ROTL}(a_{i-1},5)$=?

∇0=***+***+*…	∇R0=*+*…***+*	$=2^{30}+2^{1}$
∇1=***+*+-*…	∇R1=+-*…***+*	≅ Δ_+R0
∇2=***++--*…	∇R2=--*…***++	$=2^{30}+2^{1}+2^{0}$
∇3=**+*---*…	∇R3=--*…**+*-	≅ Δ_+R2
∇4=**+-**+*…	∇R4=*+*…**+-*	≅ Δ_+R0
∇5=**+-*+-*…	∇R5=+-*…**+-*	≅ Δ_+R0
∇6=**+-+--*…	∇R6=--*…**+-+	≅ Δ_+R2
∇7=*+--**+*…	∇R7=*+*…*+--*	≅ Δ_+R0
∇8=*+--*+-*…	∇R8=+-*…*+--*	≅ Δ_+R0
∇9=*+--+--*…	∇R9=--*…*+--+	≅ Δ_+R2
∇A=*+-*---*…	∇RA=--*…*+-*-	≅ Δ_+R2
∇B=+---**+*…	∇RB=*+*…+---*	≅ Δ_+R0
∇C=----**+*…	∇RC=*+*…----*	$=2^{30}-2^{4}-2^{3}-2^{2}-2^{1}$
∇D=+---*+-*…	∇RD=+-*…+---*	≅ Δ_+R0
∇E=----*+-*…	∇RE=+-*…----*	≅ Δ_+RC etc



$\Delta_+ m_{i-1}$

Choose signs ☑

$\Delta_\oplus m_{i-1}$

Message Expansion

$\Delta_\oplus M$

∇e_{i-1}

∇e_i

∇d_{i-1}

∇d_i

?

∇c_{i-1}

ROTL_{30}

$\nabla c_i = \mathrm{ROTL}(\nabla b_{i-1}, 30)$

$\Delta_+ f_i$?

∇b_{i-1}

∇b_i

Choose dyn. Bit pos. ☑

$\Delta_+ r_{i-1}$

ROTL_5

$\Delta_+ a_{i-1}$ $(\Delta_+ r_{i-1})$

Choose using Dyn. bit pos. ☑

$\Delta_+ a_i$ $(\Delta_+ r_i)$

$\Delta_+ a_i$

IF function

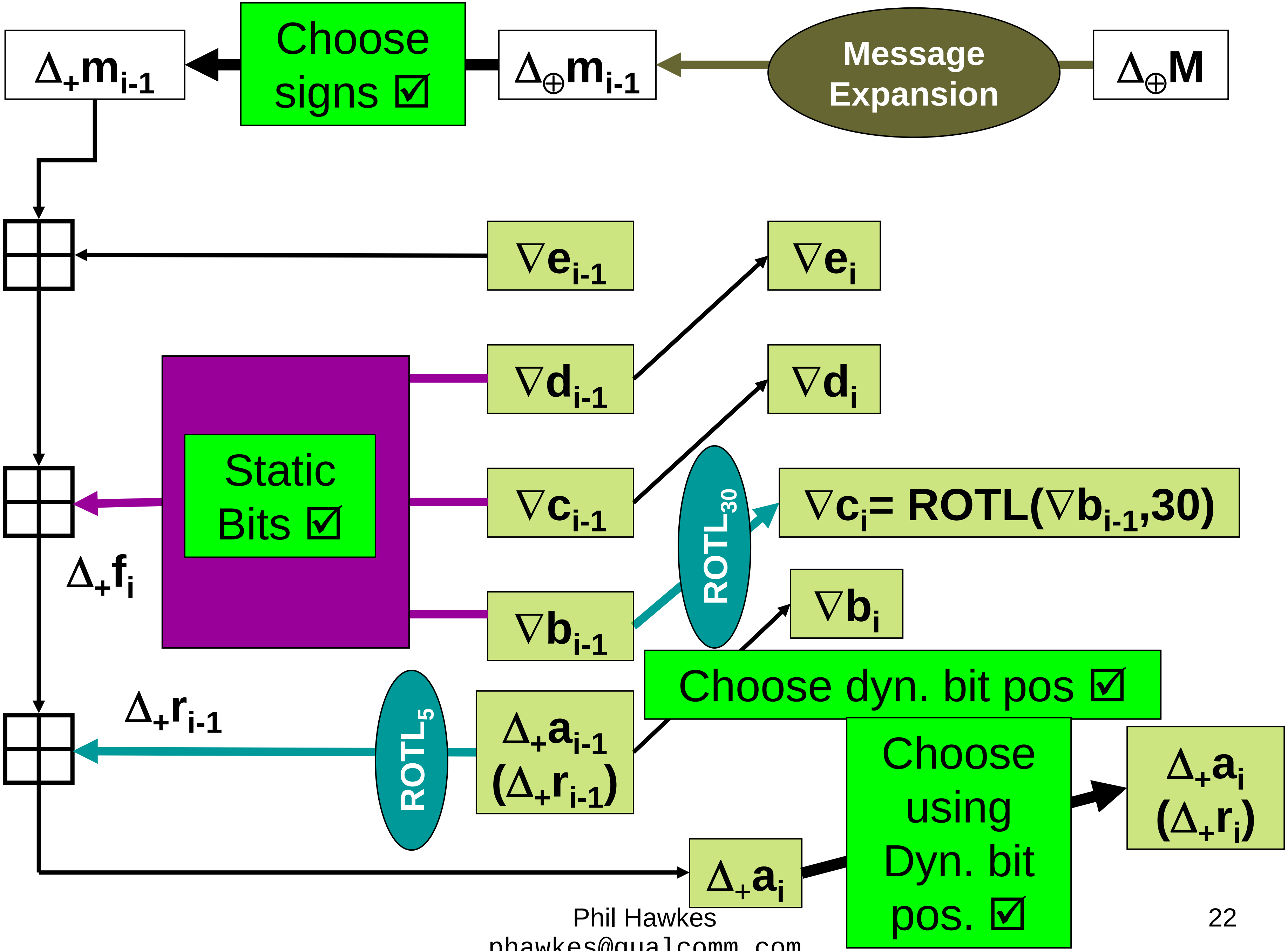
- What is known about inputs
  - Position of Dynamic & Static bits
  - Sign of Dynamic bits
- Static bits are left to specify
  - Initially ∇**b**[j]=‘*****’
  - Assign values {0,1} to static bits of **b**[j],**c**[j],**d**[j]
  - static bits of **c**[j] and ∇**d**[j] may have been assigned earlier

b[j] is Static

b	**c**	**d**	**f**	Options
Static	Static	Static	Static	
Static	Static	**Dyn.**	**Dyn.**	**b**=0
			Static	**b**=1
Static	**Dyn.**	Static	Stat	**b**=0
			Dyn.	**b**=1
Static	**Dyn.**	**Dyn.**	**Dyn.**	**b**∈{***,0,1**}

b[j] is Dynamic

b	**c**	**d**	**f**		Options
Dyn.	Static	Static	Static		**c = d**
			Dyn.	**{+, -}**	(**c,d**)∈{ (0,1),(1,0) }
				@	**c ≠ d**
Dyn.	Static	**Dyn.**	Static/**Dyn.**		**c**∈{**0,1**}
Dyn.	**Dyn.**	Static	Static/**Dyn.**		**d**∈{**0,1**}
Dyn.	**Dyn.**	**Dyn.**	Static/**Dyn.**		▨



$\Delta_+ m_{i-1}$

Choose signs ☑

$\Delta_\oplus m_{i-1}$

Message Expansion

$\Delta_\oplus M$

∇e_{i-1}

∇e_i

∇d_{i-1}

∇d_i

Static Bits ☑

∇c_{i-1}

ROTL_{30}

$\nabla c_i = \mathrm{ROTL}(\nabla b_{i-1}, 30)$

$\Delta_+ f_i$

∇b_{i-1}

∇b_i

Choose dyn. bit pos ☑

$\Delta_+ r_{i-1}$

ROTL_5

$\Delta_+ a_{i-1}$ $(\Delta_+ r_{i-1})$

Choose using Dyn. bit pos. ☑

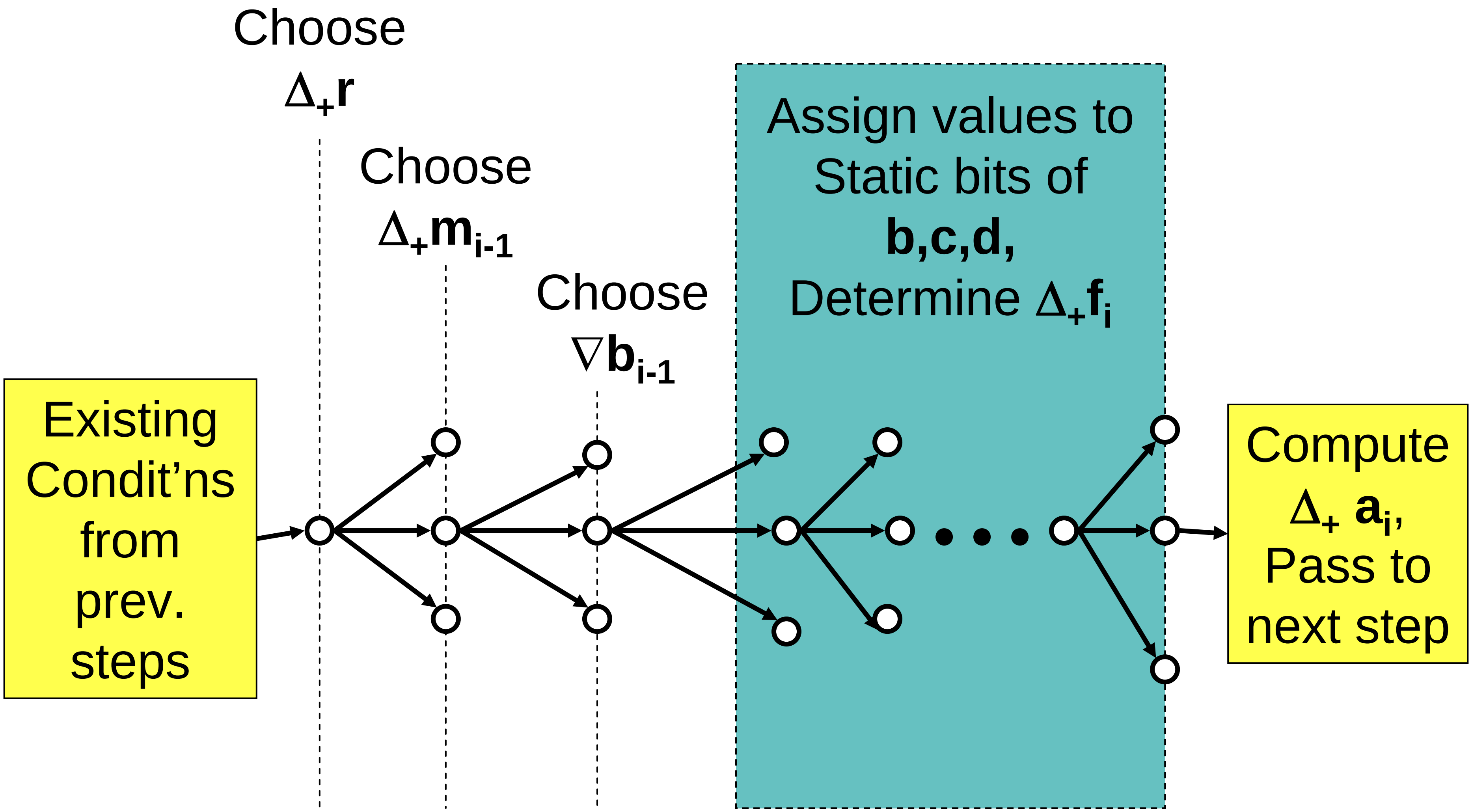
$\Delta_+ a_i$ $(\Delta_+ r_i)$

$\Delta_+ a_i$

Options at Branching Points

Know:	Want:	Fn	Choice:
$\Delta_+ a_{i-1}$	$\Delta_+ r_{i-1}$	ROTL	Positions of Dynamic bits
$\Delta_\oplus m_{i-1}$	$\Delta_+ m_{i-1}$		“Sign” of dynamic bits {+,-}
$\Delta_+ a_{i-2}$, $\Delta_+ r_{i-2}$	∇b_{i-1}		Positions of Dynamic bits
$\nabla b, \nabla c$, ∇d	$\Delta_+ f$	IF	Values of Static Bits

Branching within Forward Step



Choose $\Delta_+\mathbf{r}$

Choose $\Delta_+\mathbf{m}_{i-1}$

Choose $\nabla\mathbf{b}_{i-1}$

Assign values to Static bits of **b,c,d,** Determine $\Delta_+\mathbf{f}_i$

Existing Condit’ns from prev. steps

Compute Δ_+ $\mathbf{a}_i$, Pass to next step

Progress

- Implemented Forward search and Reverse search
- Designed comparison/matching
  - – Not implemented at time of writing