ECIT | The Institute of Electronics, Communications and Information Technology

Queen's University Belfast

Low-cost SHA-1 Hash Function Architecture for RFID Tags

Dr. Máire O'Neill (nee McLoone)

Project supported by the EU Programme for Peace and Reconciliation

Invest Northern Ireland

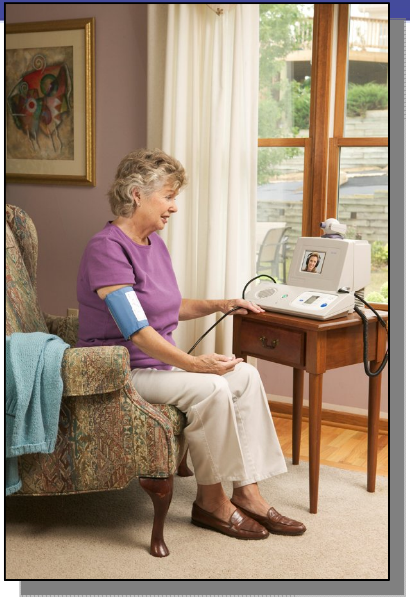
Department for Employment and Learning www.delni.gov.uk

Outline of talk

- Importance of Security in RFID applications
- Need for research into low-cost hash functions
- SHA-1 Hash Function
- Low-cost 8-bit SHA-1 hardware architecture
- Performance Evaluation
- Suitability for RFID Tags
- Conclusions

ECIT

Importance of Security in RFID



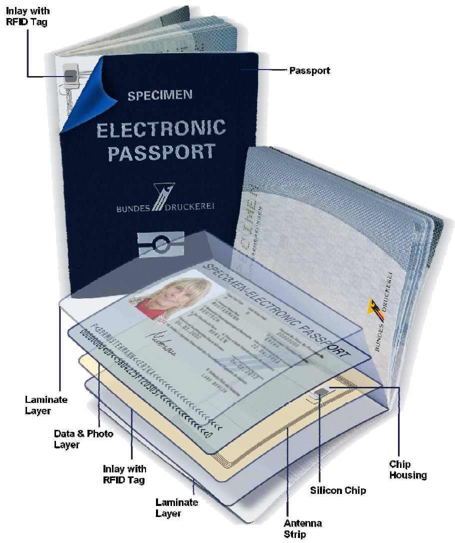
www.ce.org

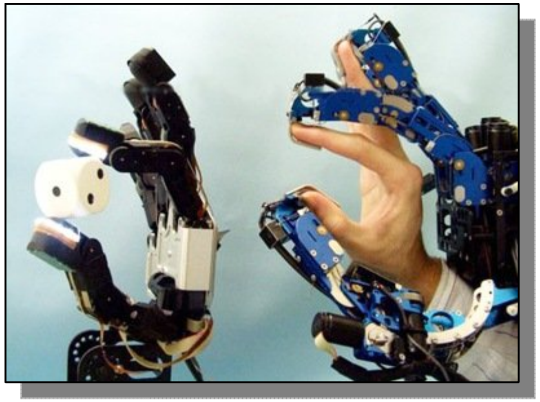
Data Security

is VITAL for emerging

Mobile & Ubiquitous

Applications





ECIT | Importance of Security in RFID

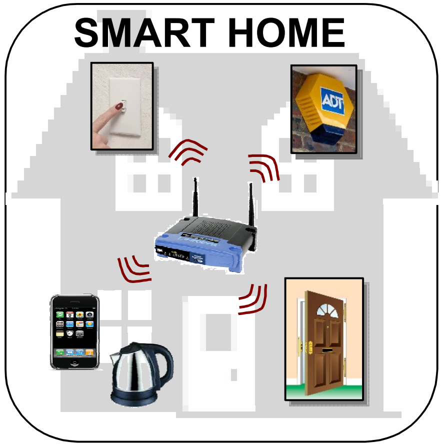
- RFID tags will play a key role in future of mobile and ubiquitous computing

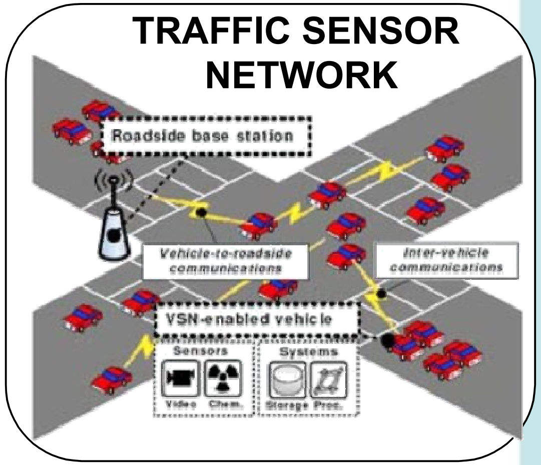
- Feb 2008 - EC Draft Recommendation on RFID Privacy and Security stated that:

  “RFID applications need to operate in a secure manner” …

  and

  “Security and privacy by design is important in the early stage of development of RFID applications”


SMART HOME


TRAFFIC SENSOR NETWORK

ECIT | Hash Functions

- Hash functions provide data integrity
- When used with digital signature algorithms & MACs they can provide authentication
- A security level of 80-bits deemed adequate for RFID tags
- Hash function with output ≥ 160-bits needed to provide RFID tag security
- Propose to use SHA-1 hash function
- Although weaknesses discovered in SHA-1 – only 2^{69} operations required to find a collision (Wang *et al.*,05) – RFID tag security may not require collision resistance

Need for low-cost Hash Functions

- Previous research into low-cost SHA-1 designs:

Design	**Area (gates)**	**Power**
Feldhofer & Rechberger	8120	35.24 μW @ 100kHz
Kaps & Sunar (*partial design*)	4276	26.73 μW @ 500kHz
Satoh & Inoue	7971	None provided
Choi *et al.*	10641	19.5 μW @ 100kHz

- Area of these designs still too large to provide security in current RFID tags

ECIT | Need for low-cost Hash Functions

- Low-cost designs of digital signature algorithms that incorporate SHA-1 have also been investigated:

  eg. ECDSA, EC Optimal El Gamal Signature scheme

- Many new security protocols proposed for RFID applications include hash functions

- Therefore, research is required into the design of:
  - New low-cost hash function algorithms;
  - Highly optimised architectures of existing hash functions

SHA-1 Hash Function

- SHA-1 was proposed by the US NIST in 1995
- Operates on a message of length $<2^{64}$ in 512-bit blocks
- Produces a 160-bit message digest
- Comprises 3 steps:
  - Message pre-processing
  - Message schedule
  - Hash computation

ECIT | SHA-1 Hash Function

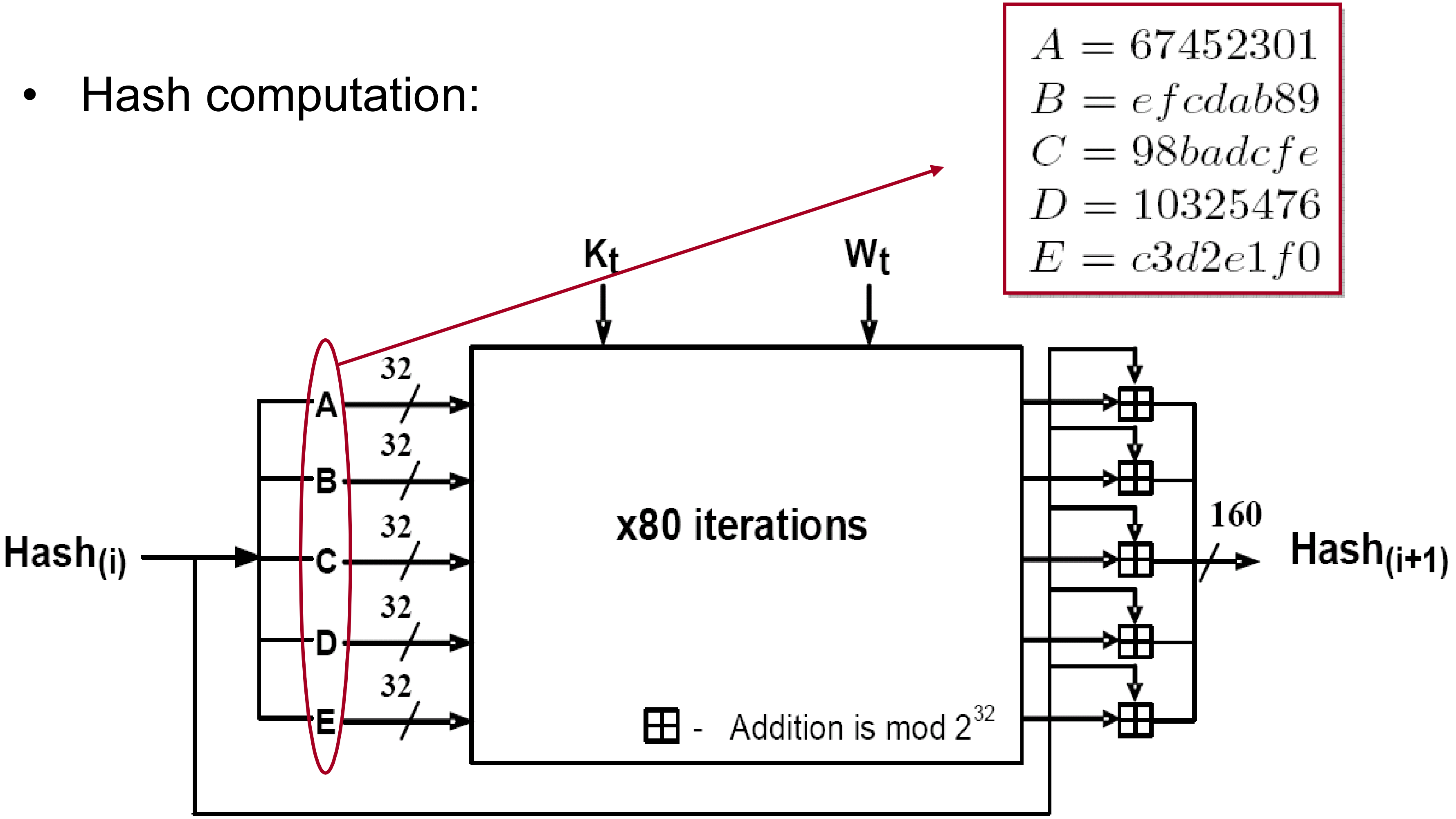
- Message pre-processing:
  - Padding the message to a length ≡ 448 mod 512;
  - Appending the message length as a 64-bit number;
  - Parsing the padded message into *N* 512-bit data blocks

- Message schedule:
  - Generation of 80 32-bit values, W_t :

$$W_t = \begin{cases} Message_t & 0 \leq t \leq 15 \\ ROT_{LEFT_1}(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) & 16 \leq t \leq 79 \end{cases}$$

SHA-1 Hash Function

- Hash computation:



$$A = 67452301$$
$$B = efcdab89$$
$$C = 98badcfe$$
$$D = 10325476$$
$$E = c3d2e1f0$$

SHA-1 Hash Function

- Hash computation:

$$
\begin{aligned}
T &= ROT_{LEFT_5}(a) + F_t(b,c,d) + e + K_t + \underline{W_t} \\
e &= d \\
d &= c \\
c &= ROT_{LEFT_{30}}(b) \\
b &= a \\
a &= T
\end{aligned}
$$

$$
\begin{array}{ll}
K_t = 5a827999 & 0 \le t \le 19 \\
K_t = 6ed9eba1 & 20 \le t \le 39 \\
K_t = 8f1bbcdc & 40 \le t \le 59 \\
K_t = ca62c1d6 & 60 \le t \le 79
\end{array}
$$

$$
F_t(b,c,d) = \begin{cases}
(b\ AND\ c)\ OR\ (\bar{b}\ AND\ d) & 0 \le t \le 19 \\
b \oplus\ c \oplus\ d & 20 \le t \le 39 \\
(b\ AND\ c)\ OR\ (b\ AND\ d)\ OR\ (c\ AND\ d) & 40 \le t \le 59 \\
b \oplus\ c \oplus\ d & 60 \le t \le 79
\end{cases}
$$

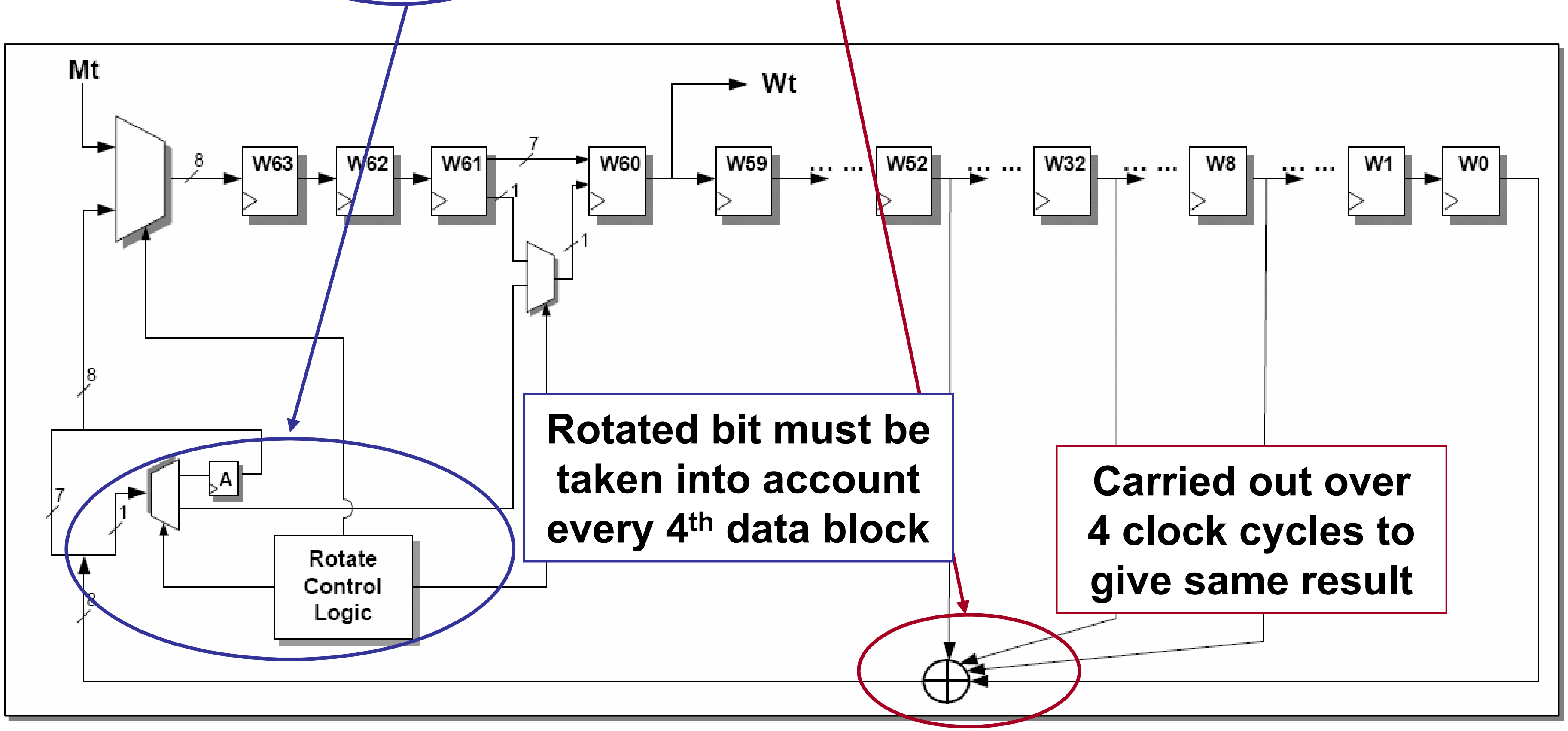
Low-cost 8-bit SHA-1 Architecture

- SHA-1 algorithm intrinsically designed to be implemented on a 32-bit platform:
  - Logical function, F_t, and rotate functions operate on 32-bit words
  - Addition is performed modulo 2^{32}
- All previous research into low-cost SHA-1 designs have employed a 32-bit architecture
- Proposal is to design an 8-bit low-cost SHA-1 architecture => modify the 32-bit oriented operations to be performed in 8-bit blocks

8-bit SHA-1 Message Schedule

- For 8-bit design, SHA-1 Message Schedule can be rewritten as:

$$W_t = \begin{cases} Message_t & 0 \leq t \leq 63 \\ ROT_{LEFT_1}(W_{t-12} \oplus W_{t-32} \oplus W_{t-56} \oplus W_{t-64}) & 64 \leq t \leq 319 \end{cases}$$



Rotated bit must be taken into account every 4th data block

Carried out over 4 clock cycles to give same result

8-bit SHA-1 Message Schedule

32-bit input, *x* = 11100001 01100010 01100011 01100100

$ROT_{left1}(x)$ = 11000010 11000100 11000110 11001001

x3 x2 x1 x0

Cycle	Input 0 Register *A*	*W63*	*W62*	*W61*	*W60/* Output
1	*x*0: 0110 0100	1100 1000	1100 1000	1100 1000	1100 1001
2	*x*1: 0110 0011	1100 0110	1100 0110	1100 0110	1100 0110
3	*x*2: 0110 0010	1100 0100	1100 0100	1100 0100	1100 0100
4	*x*3: 1110 0001	1100 0010	1100 0010	1100 0010	1100 0010

8-bit SHA-1 Hash Computation

- 32-bit *a* to *e* considered as 8-bit *a0, a1, a2, a3* to *e0, e1, e2, e3*

- Hash computation:

$$
\begin{aligned}
T &= ROT_{LEFT_5}(a) + F_t(b, c, d) + e + K_t + W_t \\
e &= d \\
d &= c \\
c &= ROT_{LEFT_{30}}(b) \\
b &= a \\
a &= T
\end{aligned}
$$

Reducing to 8-bit blocks over 4 clock cycles has no effect on equivalent 32-bit result

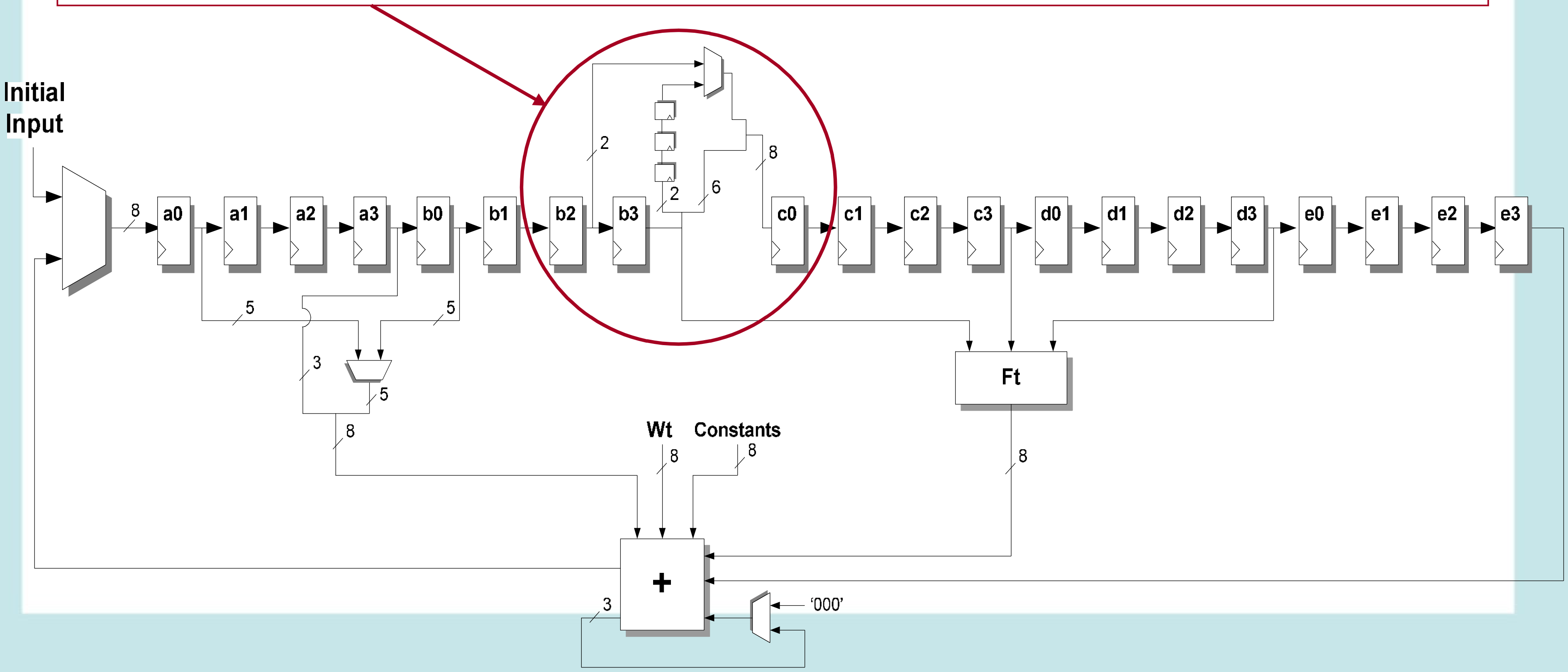
8-bit SHA-1 Hash Computation

$ROT_{LEFT_{30}}(b) \equiv ROT_{RIGHT_2}(b)$

Cycle 1: two LSBs of *b3* i/p into registers

Cycles 1, 2, 3: *c0* = two LSBs of *b2* & six MSBs of *b3*

Cycle 4: *c0* = o/p of registers & six MSBs of *b3*



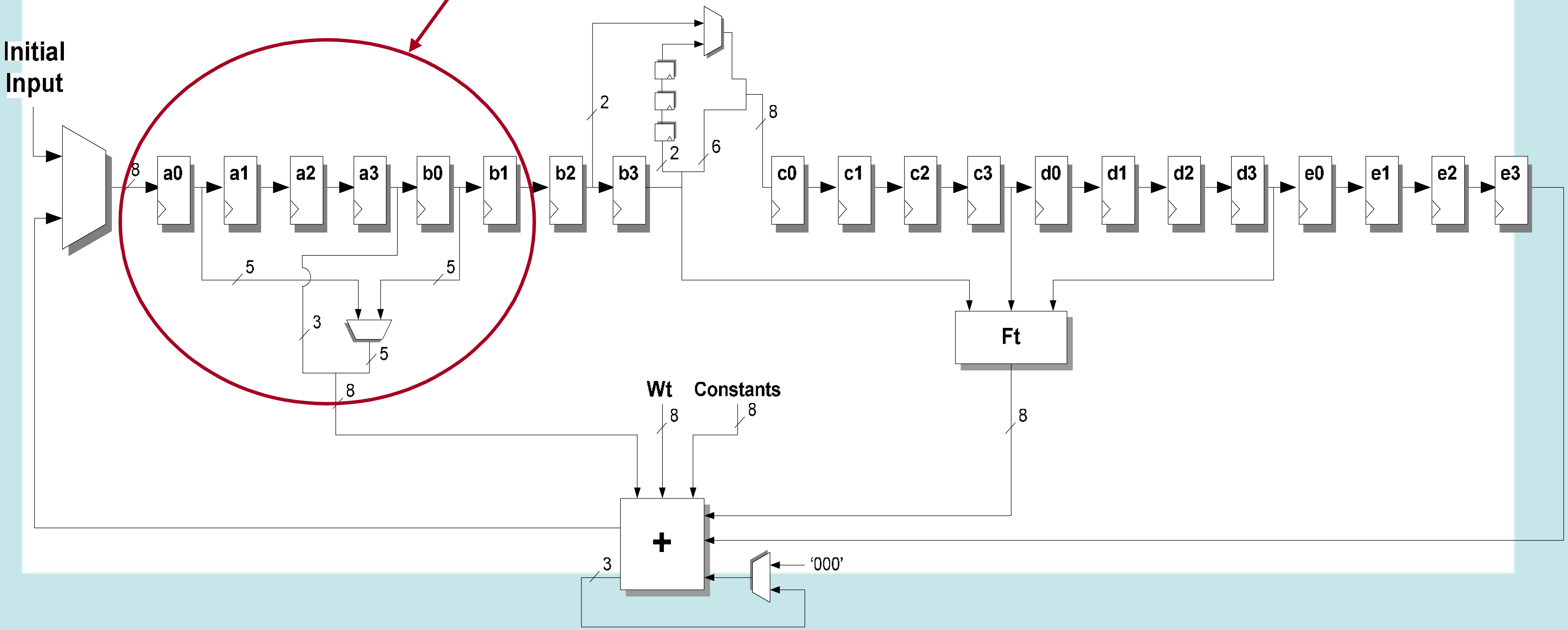
ECIT

8-bit SHA-1 Hash Computation

$ROT_{LEFT_5}(a)$ does not require any additional registers

Cycle 1: three LSBs of *a3* & five MSBs of *a0*

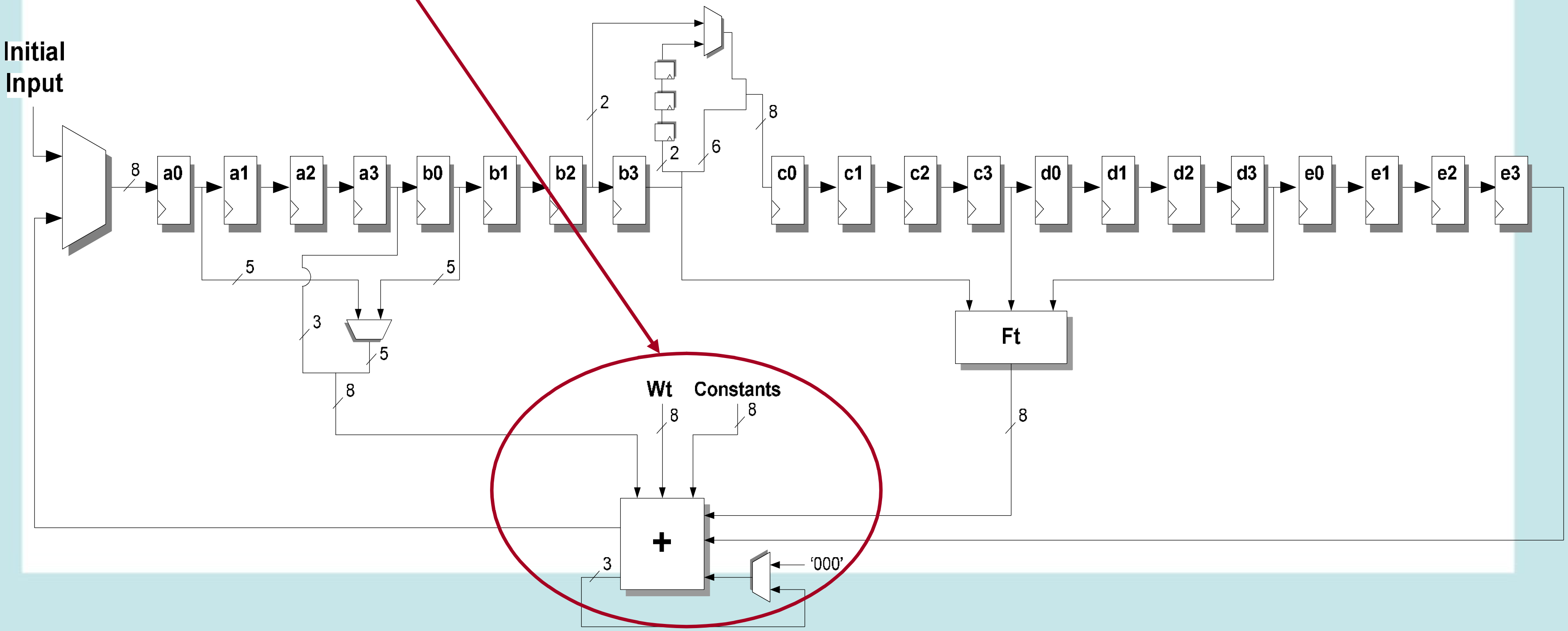
Cycles 2, 3, 4: three LSBs of *a3* & five MSBs of *b0*



8-bit SHA-1 Hash Computation

$$T = ROT_{LEFT_5}(a) + F_t(b,c,d) + e + K_t + W_t$$

Addition in 8-bit blocks requires an appropriate carry to ensure result matches equivalent 32-bit addition



ECIT

8-bit SHA-1 Final Addition

- To perform final addition:
  - As initial 8-bit *a0, a1, a2, a3* to *e0, e1, e2, e3* are input into shift register array they are also stored in a register array memory
  - After 80 iterations, o/p of register *e3* added to 8-bit register array output (carry used to ensure correct result)
- Cycle count:
  - 8-bit hash computation takes 320 clock cycles
  - Output of message schedule not available for 4 cycles
  - 160-bit hash result is output in 8-bit blocks over 20 cycles
- => Overall 8-bit SHA-1 architecture takes 344 clock cycles

Performance Evaluation

- Implementation: Faraday UMC 180nm & 130nm CMOS libs
- Synthesised: Synopsys Physical Compiler
- Power consumption: Synopsys PrimeTime PX
- Power consumption calculated as:

$$P_{max} = P_{ave} + 2 * StdDev$$

for a set of randomly generated input values

Performance Evaluation

Area utilised by 8-bit SHA-1 Architecture

Component	**Area** **0.13 µm (gates)**	**Area** **0.18 µm (gates)**
Hash computation	2751	3160
Message schedule	2655	2831
Control logic	121	131
Total	**5527**	**6122**

Note: Area can vary by ≈10% across different technologies

Performance Evaluation

Comparison with previous research

Design	Area (gates)	Power Cons (uW) @100 kHz	Timing (cycles)
This work: 8-bit SHA-1 (0.13um/1.2V)	5527	2.32	344
This work: 8-bit SHA-1 (0.18um/1.8V)	6122	13.6	344
Feldhofer & Rechberger (0.35um/3.3V)	8120	35.24	1274
Kaps and Sunar (0.13um/1.2V)	4276 (partial design)	26.73 @500 kHz	405
Choi *et al.* (0.25um)	10641	19.5	330
Satoh & Inoue (0.13um)	7971	None provided	85

Suitability for RFID Tags

- RFID Tag Limitations:

Area	Power		Timing
≈ 3000 gates	18 μW (*0.13μm/1.2V*)	27 μW (*0.18μm/1.8V*)	1800 clock cycles (*interleaved protocol*)

- Proposed 8-bit SHA-1 design meets power & timing & is within reach of area limitations
- In RFID tags, silicon area significantly impacts cost => security design area overhead must be kept to minimum
- Proposed architecture is *smallest* full SHA-1 design to date
- 8-bit design methodology gives *overall area saving of 1200 gates*

Conclusions

- Hash functions play important role in providing data security
- Will continue to be required in future ubiquitous applications
- Although SHA-1 considered weak in comparison to other hash functions, may remain suitable for some RFID applications
- Proposed a low-cost SHA-1 design based on 8-bit data path
  - 32-bit XOR, AND, NOT & OR → 8-bit functions: no logic overhead
  - 32-bit addition modulo 2^{32} & rotate functions → 8-bit functions: minimal control logic overhead
- Results in smallest SHA-1 design reported to date
- Meets RFID tag power and timing limitations & is within reach of area requirement