ROM Design and Evaluation against Power Analysis Attack

Huiyun Li, Simon Moore
Computer Laboratory, University of Cambridge
Firstname.Surname@cl.cam.ac.uk

Abstract

Memories are crucial components in smart cards to store operating system routines, secret key information, or data being computed. For example, the differential power analysis (DPA) attack on the DES algorithm generally focuses on S-boxes. Designing and evaluating memories is therefore an important task in smart card design. As a case study, the power consumption of a normal 8×8 Read-only Memory(ROM) is simulated in HSPICE. Randomness is later inserted to mask the data-dependent information leakage. A dual-rail version of the ROM is then presented and appears to be a better countermeasure against power analysis attack. The data-dependent information leakage of all models is evaluated quantitatively with the correlation coefficient between the ROM's Hamming weight and power consumption.

1 Introduction

1.1 Power simulation on an 8×8 ROM

The ROM is designed to be 3-bit input, 8-bit output as shown in Figure 1. It consists of two main components: a 3-to-8 decoder and a memory array. The decoder is made up of eight 3-input `AND` gates each driven by a min-term of the 3 input signals. The memory array is an array of pull-down N-type transistors, on each intersection of a horizontal address line and a vertical data line.

A HSPICE netlist of transistors with RC wire model is used to simulate power of this simple ROM design. We increase the Hamming weight (the number of "1"s) of the ROM content one by one. For each Hamming weight, we randomly distribute the locations of "1"s (N-type transistors) and run power simulations around 10 times. The power consumption versus Hamming weight graph shown in Figure 2 demonstrates that Hamming weight information is leaked, as average power increases linearly with it.

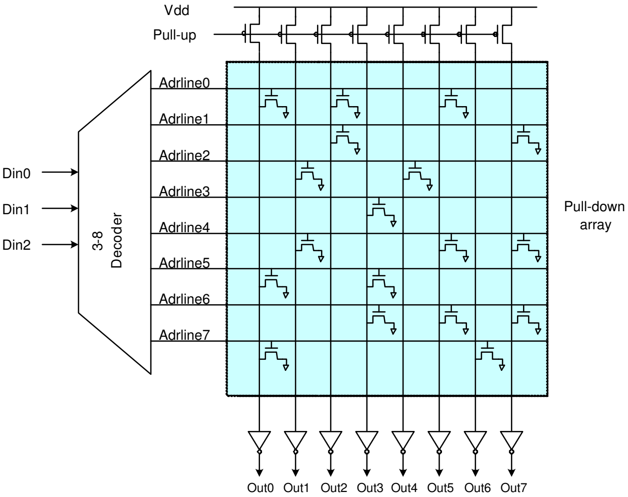

Figure 1: 8×8 ROM

2 Inserting randomness into ROM

There are two dimensions of freedom which cause power consumption variation given a certain Hamming weight:

- Duty cycle of address lines
- N-type transistor distribution

The *duty cycle* of address lines are not identical to each other, due to inverter delay in the address

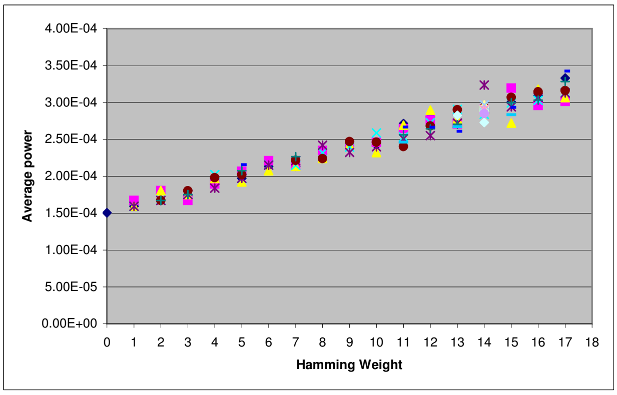

Figure 2: Increased average power over increasing Hamming weight

decoder. When one address line is selected and the N-type transistors on it are turned on, the power dissipation caused by short-circuit current is approximately proportional to the duty cycle of selected address line. As a result, the power consumption differs when locations of N-type transistors change between different address lines.

The power consumption variation caused by duty cycle nuance can be exploited to mask the linearity between the power and the Hamming weight. One may consider increasing the duty cycle nuances in address lines. But the influence would be slight since differences of some duty cycles is very small. Moreover, it increases the risk of timing analysis attack which in turn cancels the improvement on power information leakage.

An alternative is to modify the *N-type transistor distribution* by using extra dummy bitlines, i.e. to increase the scope of N-type transistor distribution over a larger ROM whose circuit is shown in Figure 3. We run power simulation on the randomness inserted ROM, and discover this technique effectively obscures the Hamming weight information. The power consumption variation for each Hamming weight is increased as illustrated in Figure 4. To obtain same amount of useful information demands more samples now to average out randomness. This successfully raises the time penalty of the power analysis attack.

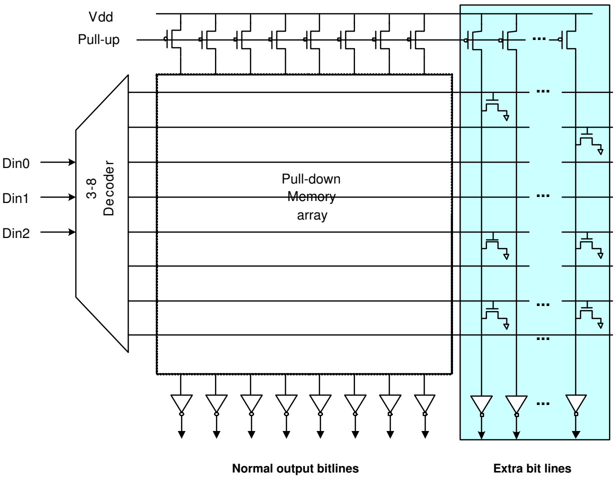

Figure 3: 8 × 8 ROM with extra bitlines, for randomness insertion

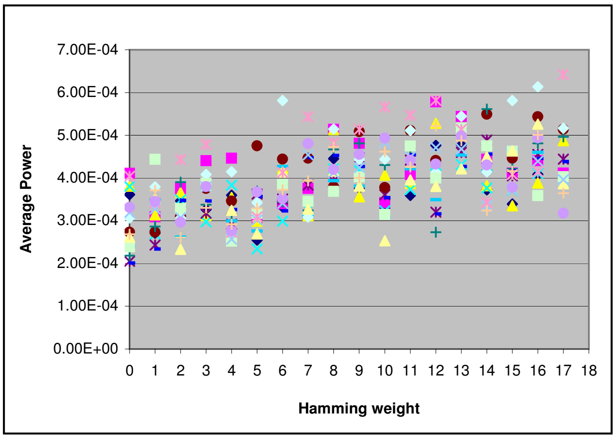

Figure 4: Average power over increasing Hamming weight, increased power consumption variation when randomness inserted

3 Dual-rail ROM design and power simulation

Power simulation on a longer range of Hamming weights (from 1 to 64 in Figure 5, compared to 1 to 18 in Figure 2) indicates linearity is still observable. We reckon then a dual-rail ROM design may be a better countermeasure. *Dual-rail* refers to an encoding system where two-bit value “01” stands for *logic-0*, “10” for *logic-1*. The dual-rail ROM has a double number of bitlines, which in pairs represent logic words. With this encoding technique, a constant number (half the number of bitlines) of N-type transistors will be turned on no matter which address line is selected.

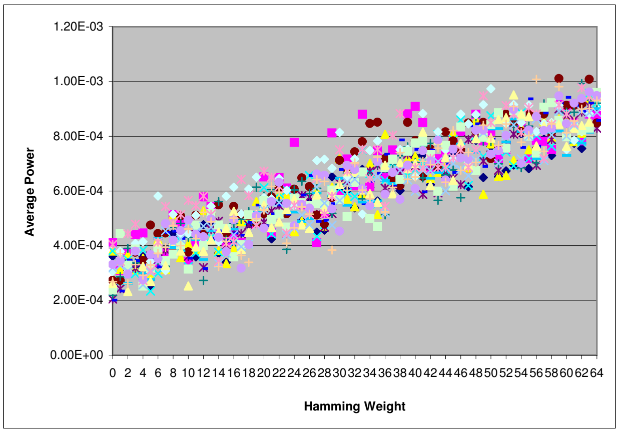

Figure 5: Average power of bundle-data ROM over increasing Hamming weight, linearly increased power still observed even with randomness insertion

Figure 6 shows a dual-rail 8 × 8 ROM example which has 16 bitlines to output 8-bit words. We run power simulation on it similar to its bundle-data version, but increasing the number of *logic-1* instead of increasing the number of “1”s, which is consistently equal to half of the total intersections of address lines and bitlines. The flat narrow power pattern in Figure 7 depicts that constant power is consumed regardless of ROM content (logically). The energy dissipated is equal to that of the bundle-data version when Hamming weight is 64.

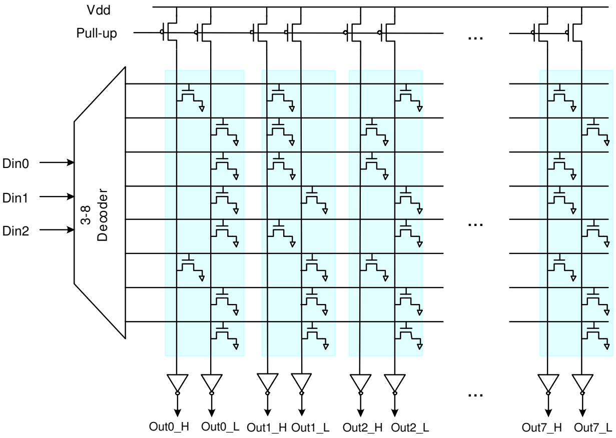

Figure 6: Dual-rail 8 × 8 ROM, 16 bitlines representing 8-bit word.

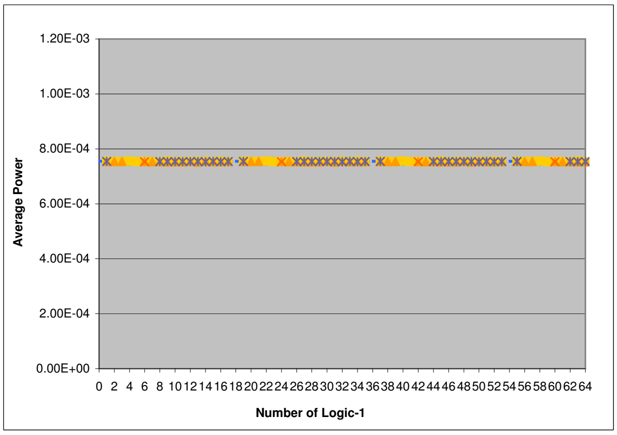

Figure 7: Average power of dual-rail ROM over increasing number of *Logic-1*, as a comparison of Figure 5

4 Analysis with correlation coefficient

Correlation coefficient is a statistic measure of the relation between two or more variables, obtained by dividing their covariance over individual standard deviations. The value can range from -1.00 to +1.00. The absolute value of 1.00 represents a perfect correlation and a value of 0.00 represents a lack of correlation. Correlation coefficient is usually used to test the linearity between two variables. However, we use it here to evaluate the correlation between Hamming weight and power consumption for a certain circuit. For the above 8 × 8 ROM design, the corresponding correlation coefficient is estimated by the following formula, where W denotes power consumption, H denotes Hamming weight of the ROM content.

$$\begin{aligned}\rho_{WH} &= \frac{cov(W,H)}{\sigma_W \sigma_H} \\ &= \frac{\sum_{i=1}^{n}(W_i - \overline{W})(H_i - \overline{H})}{\sqrt{\sum_{i=1}^{n}(W_i - \overline{W})^2 \cdot \sum_{i=1}^{n}(H_i - \overline{H})^2}}\end{aligned}$$

Since the value of correlation coefficient largely relies on the sample size, we fix the Hamming weight interval (from 0 to 64) and run 10 power measurements for each Hamming weight. The following table shows the correlation coefficients for the three ROM models: the normal bundle-data ROM, randomness inserted bundle-data ROM and the dual-rail ROM. It proves our intuition that a dual-rail ROM can provide much lower linearity and be a better defence against power analysis attack.

Table 1: Correlation coefficients (ρ_{WH})for ROM models

ROM type	ρ_{WH}
normal bd_ROM	99%
randomness inserted bd_ROM	92%
dr_ROM	47%

5 Conclusion

Power analysis has been simulated on a 8 × 8 Read-only Memory(ROM) in HSPICE. A high correlation between power consumption and Hamming weight is illustrated and can be exploited to guess ROM content. We therefore insert randomness with extra dummy address lines, which helped to reduce linearity from 99% to 92%. To further mask the data-dependent information leakage, we propose a dual-rail ROM, where same amount of N-type transistors are conducted regardless of the ROM content. This dual-rail ROM model has demonstrated to be data-independent and has achieved linearity as low as 47%. We believe the dual-rail technique can be used for EEPROM or SRAM in smart cards to offer them robustness against memory readout by power analysis.

References

[1] K. Tiri, M. Akmal and I. Verbauwhede, "A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards", *Proc. IEEE 28th European Solid-state Circuit Conf. (ESSCIRC'02)*, 2002

[2] Star-Hspice manual, Avant!, 1999.

[3] D. Samyde, S. Skorobogatov, R. Anderson, and J.-J. Quishquater, "On a new way to read data from memory." http://www.cl.cam.ac.uk/ftp/users/rja14/SISW02.pdf, 2002.