<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>An energy-efficient 2.5D through-silicon interposer I/O with self-adaptive adjustment of output-voltage swing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Xu, Dongjun; Sai Manoj, P. D.; Huang, Hantao; Yu, Ningmei; Yu, Hao</td>
</tr>
<tr>
<td><strong>Citation</strong></td>
<td>Xu, D., Sai Manoj, P. D., Huang, H., Yu, N., &amp; Yu, H. (2014). An energy-efficient 2.5D through-silicon interposer I/O with self-adaptive adjustment of output-voltage swing. Proceedings of the 2014 International symposium on low power electronics and design, 93-98.</td>
</tr>
<tr>
<td><strong>Date</strong></td>
<td>2014</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://hdl.handle.net/10220/25748">http://hdl.handle.net/10220/25748</a></td>
</tr>
<tr>
<td><strong>Rights</strong></td>
<td>© 2014 Association for Computing Machinery. This is the author created version of a work that has been peer reviewed and accepted for publication by Proceedings of the 2014 international symposium on Low power electronics and design, Association for Computing Machinery. It incorporates referee’s comments but changes resulting from the publishing process, such as copyediting, structural formatting, may not be reflected in this document. The published version is available at: [<a href="http://dx.doi.org/10.1145/2627369.2627614">http://dx.doi.org/10.1145/2627369.2627614</a>].</td>
</tr>
</tbody>
</table>
ABSTRACT
A self-adaptive output swing adjustment is introduced for the design of energy-efficient 2.5D through-silicon interposer (TSI) I/Os. Instead of transmitting signal with large voltage swing, Q-learning based self-adaptive adjustment is deployed to adjust I/O output-voltage swing under constraints of both power budget and bit error rate (BER). Experimental results show that the adaptive 2.5D TSI I/Os designed in 65nm CMOS can achieve an average of 13mW I/O power, 4GHz bandwidth and 3.25pJ/bit energy efficiency for one channel under 10^-6 BER, which has 21.42% reduction of power and 14.47% energy efficiency improvement.

Categories and Subject Descriptors: B.4.2 [Input/Output Devices]:Channels and controllers

Keywords: output-voltage swing tuning; TSI I/O; 2.5D integration; I/O Channel controller; Q-learning.

1. INTRODUCTION
There is an emerging need to process large amount of data with high bandwidth and low power consumption I/Os for an energy-efficient cloud-server, which is mainly based on the integration of many-core processors with shared memory [1]. A 3D integration by stacking several layers of dies vertically using through-silicon via (TSV) [2, 3, 4] has better scalability of integration but worse thermal density for heat dissipation [5, 6]. A 2.5D integration by through-silicon interposer (TSI) in common substrate has gained recent interest for memory-logic integration in cloud-server design due to better thermal dissipation capability [7, 8]. Compared to the 3D integration by TSV, TSI based 2.5D also enables the integration of transmission line (T-line) based I/O design to achieve high bandwidth and low power [9]. It has no area overhead because the interposer based T-line can be deployed underneath the substrate. Hence it has become an interest to design energy-efficient I/Os using TSI based T-line for the integration of many-core microprocessors with shared memory.

Previous work of wire-line communication by PCB trace of backplane [10] has large latency and poor signal-to-noise ratio of the transmission channel. Moreover, all previous works [11, 12] assume uniform output-voltage swing that consumes large I/O communication power. To meet a low communication power budget, output-voltage swing at transmitter can be reduced. However, the reduction in output-voltage swing increases bit error rate (BER) at receiver [13, 14]. Therefore it has become a trade-off to balance the BER and energy efficiency during the I/O communication by TSI. In this paper, we have proposed an adaptive 2.5D I/O design that can automatically adjust output-voltage swing with balanced consideration of energy efficiency and BER. An error correcting code (ECC) is developed for checking BER. One Q-learning based management is applied to adjust the level of output-voltage swing at transmitter (associated with cores) such that one can achieve a reduced power under specified BER requirement.

The adaptive I/Os are integrated with the TSI based transmission line (T-line) implemented in 65nm CMOS process. Experimental results show that the adaptive 2.5D TSI I/Os designed in 65nm CMOS can achieve an average of 13mW I/O power, 4GHz bandwidth and 3.25pJ/bit energy efficiency for one channel under 10^-6 BER, which has 21.42% reduction of power and 14.47% improvement of energy efficiency. The remainder of this paper is organized as follows. Firstly, we describe the memory-logic integration architecture by 2.5D TSI integration with an adaptive I/O design; and the according problem of adaptive control in Section 2. Section 3 presents the circuit blocks of 2.5D TSI based receiver/transmitter, error-correcting code and adaptive tuning. In Section 4, the adaptive output-voltage swing tuning by the Q-learning algorithm is presented. The experimental results are shown in Section 5 with conclusion in Section 6.

2. 2.5D TSI I/O COMMUNICATION
In this section, we will present memory-logic integration architecture by TSI I/O followed by problem formulation of the adaptive I/O design for power versus BER.

2.1 Memory-logic Integration by 2.5D TSI I/O
The traditional interconnection between processors and memories is by printed circuit board (PCB) with backplane [10] containing sockets into which other boards can be plugged in (See Fig. 1(a)(i)). However, long trace (≥ 25cm ) and non-ideal vias are needed at PCB scale, hence there is a severe loss on the backplane, which requires current-starved circuits to reach the high data rate and equalizers to compensate the channel loss [10]. For the 2.5D TSI technology [7], the processors and memories dies are integrated on one common substrate by silicon interposer underneath (See Fig. 1(a)(ii)). Unlike traditional backplane based interconnects, 2.5D TSI Ts are much shorter with a few mm in length and are deployed underneath the substrate with less routing overhead. The channel loss vs. frequency is shown in Fig. 1(b) for PCB backplane I/O and 2.5D TSI I/O, respectively. When comparing the loss at 5GHz clock frequency, the PCB backplane with long (25cm) trace has nearly 24dB channel loss; and the TSI with small trace (10mm width, 3mm length) has only 1dB loss. Hence, the 2.5D TSI based integration has much less loss with better performance for the memory-logic-integration. When compared to the 3D through-silicon-via based integration, the 2.5D TSI based integration further shows much better thermal dissipation capability [7, 8].
To achieve high bandwidth and energy efficiency, we will study 2.5D TSI I/O in this paper for the integration of multi-core microprocessors and memories. In order to improve the energy efficiency, we propose a self-adaptive design with tuning of the output-voltage swing by checking the BER and power. Compared to the previous designs [11] with fixed full output-voltage swing, the proposed design can save I/O communication power and improve energy efficiency.

Architecture for memory-logic integration by the 2.5D adaptive TSI I/O is shown in Fig. 2(a). Each of the memory and core will have transmitter as well as receiver to enable a full duplex communication. The data is encoded by adding the redundant parity check bits as input to the transmitter. Serializer converts the parallel data into serial data for transmission through the TSI channel. At the receiver end, the encoded data is decoded and BER is calculated by ECC block. One I/O controller tunes the output-voltage swing adaptively by varying the driver tail current at the transmitter. The transmitter adjusts its output-voltage swing adaptively based on the feedback of the BER and I/O communication power. Thus, the output-voltage swing can be tuned adaptively to save the power along with considering the BER constraint. Detailed description of each of the transmitter, receiver, coding and the adaptive tuning of output-voltage swing is presented in Section 3. In the following, we further show how to formulate an adaptive control problem of the proposed I/O.

2.2 Problem Formulation

Note that a large output-voltage swing may be not required when certain amount of error in the data can be accepted for the trade-off of power saving. This is particularly true for some applications involving data such as imaging, audio and video. Therefore, one can reduce the output-voltage swing at transmitter though some error may be accepted when detected at receiver. By employing the error-correcting code (ECC) block at the receiver side, one can determine the BER and check if there exists margin. Based on this phenomenon one can design a controller at transmitter to balance the trade-off between the I/O communication power and BER by tuning the voltage levels of the output-voltage swing, which can be formulated as:

Problem: Tune the output-voltage swing at the transmitter to achieve low power at the cost of BER based on the logic-memory communication characteristics.

\[
\text{Opt. } < P_i, \text{BER}_i >
\]

\[\begin{align*}
(\text{i}) \quad & P_i \leq P_B \\
(\text{ii}) \quad & \text{BER}_i \leq \text{BER}_T
\end{align*}
\]

where \(P_i\) and \(\text{BER}_i\) denotes the I/O communication power and BER under the \(i\)-th output-voltage swing level \(V_{s_i}\). Note that the BER and power are both functions of the output-voltage swing, \(P_B\) and \(\text{BER}_T\) represents the targeted I/O communication power and BER of one TSI I/O under the normal operation. With the increase in the output-voltage swing, the I/O communication power increases and BER goes down and vice-versa. Output-voltage swing level \(V_{s_i}\) needs to be adaptively tuned for optimizing the I/O communication power and BER simultaneously.

In this paper, a self-adaptive tuning of the output-voltage swing at transmitter is performed based on one Q-learning algorithm with feedback of the BER and power as inputs, presented in Section 4. In the next section, we first discuss the detailed circuit blocks of 2.5D adaptive TSI I/Os.

3. 2.5D TSI I/O CIRCUIT DESIGN

In this section, we discuss in detail about each component of the overall 2.5D TSI I/O link such as transmitter (Tx) and receiver (Rx) presented in Fig. 2(b). To operate high bandwidth by single channel of 2.5D TSI I/O, we employ 8:1 serializer in the Tx and 1:8 de-serializer at the Rx. Each of the Tx and Rx has a voltage-controlled-oscillator (VCO) to generate the required clock signal (2GHz). Both the Tx and Rx are terminated for the 2.5D TSI based T-line with matched 50Ω resistor. At the Rx, the serial bit stream is sampled and de-serialized; and is re-synchronized by the recovery clock from the clock data recovery (CDR) block.

3.1 Receiver and Transmitter

In details, the Tx employs a 8:1 serializer to convert 8-bit parallel data into serial data as shown in Fig. 3(a). Four digital D flip-flops are implemented as a shift-register chain for each of the odd \((D_1, D_3, D_5, D_7)\) and even \((D_0, D_2, D_4, D_6)\) bits of data. This is followed by a 2:1 MUX to combine them altogether. A current-mode logic (CML) output driver is used to drive the TSI T-line from the Tx to the Rx on the common substrate. The CML output stage is powered by the fixed supply (1.2V). The I/O communication power \(P\) depends on the output-voltage swing and the tail current of the driver. I/O Communication power and BER are considered as state of the system, and can be tuned by the output-voltage swing, which will be considered as the corresponding action. For example, one can generate control bits to tune the tail current of the CML driver and thus the output-voltage swing, as shown in Fig. 3(b).

What is more, compared to the traditional serial I/Os based on the backplane PCB trace [15, 16], the 2.5D TSI I/Os do not need the complex equalizer circuits at the receiver due to the small signal loss in the TSI T-line channel. A sampler at the front-end receiver is employed to convert the current-mode signals into digital levels. After the data decision, this data is processed in the digital domain, which saves more power compared to analog de-multiplexer. A delay-locked loop (DLL) based clock-data recovery...
errors. It can be used to obtain the historical BER at the receiver and is further fed back to the transmitter. BER for a certain time interval is calculated by the total number of errors found to the total number of bits transmitted.

3.3 Adaptive Tuning

Based on the calculated BER from the ECC, a feedback signal is sent back to the I/O controller at the transmitter. This signal is considered as one of the component for control that forms a look-up-table (LUT). The I/O controller generates the corresponding control bits. The control bits can control the DAC current at the tail of CML buffer driving the TSI T-line. Thus, the output-voltage swing is tuned by varying the tail current of CML buffer. As shown in Fig. 3(c), the CML driver with variable current source is set by the DAC current and load resistor. The DAC tail current source is composed of a group of current sources in parallel with switches controlled by the control bits generated from the I/O controller. Generally, the load resistor is set 50Ω for the TSI T-line impedance matching. In this paper, tail current source is varied from 2mA to 5mA.

4. Q-LEARNING BASED ADAPTIVE TUNING

In this section, we will discuss about the Q-learning theory and its application to have a self-adaptive tuning of the output-voltage swing for the 2.5D TSI I/Os.

4.1 Q-learning Theory

The Q-learning theory [17] is generally practised to find an optimal action-selection policy from the set of states $S$. The Q-learning algorithm evaluates state and action pairs form the previous inputs. To solve (1), we formulate the Q-learning algorithm that takes the I/O communication power $P_i$ and BER $BER_i$ as the state vector; and uses the output-voltage swing level $V_{out}$ as the action by

$$ S = < P_i, BER_i >. $$

In order to obtain the state and action pairs and form a look-up-table (LUT), the input samples are trained. A sample LUT will be as follows:

<table>
<thead>
<tr>
<th>Action</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{out}$</td>
<td>$P_i$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

The input samples are collected at a time periods of control cycle, in scale of $ns$. Duration of control cycle is based on the speed of I/O controller circuit. The next state variable needs to be predicted with an action for the next input sample. This can be done by calculating a reward function to achieve an optimally estimated value based on the existing state $S_k$ by

$$ R_e = f(S_{k+1}) - f(S_k). $$

$$ S_{k+1} = \frac{1}{P_i} \left( R_e + P_i \cdot BER_{k+1} \right) $$

$$ f(S_{k+1}) = f(S_k) $$

$$ f(S_{k+1}) = f(S_k) $$
Here \( S \) denotes the predicted state and \( k \) indicates the number of control cycle. The reward \( R_w \) denotes the direction of state transition. The optimal estimation is chosen among the set of states to satisfy the desired criteria by taking the corresponding action selected from the formed LUT. The optimal estimation \( E \) of the states \( S \) can be calculated as follows:

\[
E = \min\{f(S_{k+1-i})\}, i = 0, ..., M. \tag{3}
\]

Here \( M \) denotes the number of samples to make the optimal estimation. Based on the optimal estimation \( E \), the action to be taken for next state is calculated as

\[
f(S_{k+1}) = f(S_k)(1 - \alpha) + \alpha(R + \gamma E) \tag{4}
\]

where \( f(S_{k+1}) \) is the determined action for the next control cycle \( k + 1 \); \( \alpha \) denotes learning rate and \( \gamma \) denotes discount factor. In the following, we show the system power and BER models with predictions.

### 4.2 System Power Model

The first component of the state vector is the I/O communication power. The system power model refers to the I/O communication power of driver and the TSI T-line power, both depending on the output-voltage swing \( V_s \). For the CML based driver with TSI T-line [18] the I/O communication power is given by

\[
P_i = V_s t_i (1 + \eta * V_{dd} * s \frac{R_D + Z_{diff}}{s} * f). \tag{5}
\]

Here \( I_t \) is driver tail current; \( s \) is duration of signal pulse; \( \eta \) is activity factor; \( R_D \) is the resistance of driver; and \( Z_{diff} \) is the characteristic impedance of the TSI T-line.

The tail current \( I_t \) at the current control cycle can be obtained from measurement and is predicted for the next control cycle by auto-regression (AR)

\[
I_t(k + 1) = \sum_{i=0}^{M-1} w_i I_t(k - i) + \xi. \tag{6}
\]

Here \( I_t(k + 1) \) denotes the predicted tail current at \( k + 1 \)-th control cycle; \( w_i \) represents the auto-regression coefficient; \( \xi \) is the prediction error and \( M \) represents the order of the AR prediction. Based on the predicted tail current, the I/O communication power for next control cycle is calculated as \( P_i(k + 1) \).

### 4.3 System BER Model

The second component of the state vector is the BER, the feedback from the receiver. The BER is affected by the output-voltage swing, external noise, channel noise etc. [19]. In a wireline communication system [20], the BER has a relationship with the output-voltage swing as follows

\[
BER_i = \frac{1}{2}\text{erfc}(\frac{V_s}{\sqrt{2} \sigma_v}). \tag{7}
\]

Here erfc is the complementary error function and \( \sigma_v \) is the standard deviation of the noise.

The BER can be obtained as the feedback from ECC at the receiver. During the training process, \( \sigma_v \) is calculated from (7) for the given output-voltage swing at the current control cycle. By knowing \( \sigma_v \), the BER for the next control cycle can be predicted accordingly.

### 4.4 Q-learning Control Flow

The self-adaptive tuning of the output-voltage swing at the CML buffer is performed based on the Q-learning discussed in Section 4.1. The I/O communication power \( P_i \) and BER \( BER_i \) are considered as the state vectors with output-voltage swing level \( V_s \) as the corresponding action.

The proposed self-adaptive output-voltage swing tuning is presented in Algorithm 1. LUT is formed with output-voltage swing as action based on the I/O communication power and BER in (5) and (7) as state vectors. The tail current of the CML buffer is predicted, as given in (6), Line 1 of Algorithm 1. Based on the predicted tail current, the I/O communication power for the control cycle is calculated using (5). Similarly, BER is predicted using the feedbacked \( \sigma_v \).

Based on the present and predicted power and BER values, reward \( R_w \) is calculated similarly to (2). Since we have two factors, we consider the weighted sum, as given in Line 3 of Algorithm 1. Here \( a_1 \) and \( a_2 \) denote the weighted coefficients for normalized rewards of the communication power \( \Delta V_{ss} (P_i) \) and normalized BER \( \Delta V_{Ber} (BER_i) \). After calculating the reward, the optimally estimated value for the output-voltage swing is calculated, as in Line 4 of Algorithm 1. Finally, the output-voltage swing is selected (4), given in Line 5 of Algorithm 1. The LUT can be implemented online with the corresponding control bits calculated and feedbacked to CML buffer to tune the DAC current of the CML buffer. This is how the adaptive tuning is performed. Note that LUT can be implemented in the hardware with multiple AND/OR partial matching logic circuit instead of read only memory (ROM). This LUT based implementation has higher speed and lower power compared to the ROM. As a summary, the whole flow of adaptive tuning by the Q-learning algorithm is shown in Fig. 6.

---

**Algorithm 1:** Q-learning based adaptive tuning of output-voltage swing

**Input:** Communication power trace \( P_i \), BER feedback from receiver and look-up-table (LUT)

**Output:** Adaptive tuning of output-voltage swing \( V_s \)

1. Predict tail current: \( I_t(k + 1) = \sum_{i=0}^{M-1} w_i I_t(k - i) + \xi \)
2. Calculate corresponding communication power and BER
3. Reward: \( R_w = a_1 \Delta V_{ss} (P_i) + a_2 \Delta V_{Ber} (BER_i) \)
4. Optimal value estimate: \( E = min\{V_s((i - j + 1))\}, j = 0, ..., M \)
5. \( V_s(t + 1) \leftarrow V_s(t)(1 - \alpha) + \alpha(R_w + \gamma E) \)
6. By adjusting tail current using control bits, tune corresponding \( V_s \)

---

**Figure 6:** Flowchart showing Q-learning based self-adaptive voltage swing tuning

**5. SIMULATION RESULTS**

The 2.5D adaptive TSI I/O verification is performed in Cadence Virtuoso (Ultrasisium-Verilog) and Matlab. An 8-core MIPS microprocessor with 8-bank of SRAM memory is designed with GF 65nm CMOS. The 2.5D TSI T-line is of length 3mm and 10µm width, driven by the CML buffer. The power traces are measured from Cadence Virtuoso and control cycle is set as \( \text{ns} \), larger than switching time of I/O controller. The look-up-table (LUT) is designed with ECC coding for adaptively tuning the output-voltage swing. The controller is based on the Q-learning of the I/O communication power and BER at receiver respectively. The multiple setup parameters are from:
(100mV, 4.98E − 2mW, 5.05E − 2), (150mV, 1.75E − 1mW, 1.10E − 3), (200mV, 2.55E − 1mW, 1.12E − 4) and (300mV, 3.48E − 1mW, 8.40E − 6). This LUT is almost robust, since this depends on characteristics of the circuit rather than the application. The other geometry settings are presented in Table 1. Area overhead and power overhead of the adaptive tuning is nearly same as that of the I/O controller, presented in Table 1. The circuit is designed in Cadence with the according technology PDK. The overall I/O performance can provide 76mV − 190mV peak-to-peak signal swing with 4Gb/s bandwidth, and the power consumption is only 13mW. The adaptive self-tuning of output-voltage swing may come with a little area overhead of 0.03mm² for additional control circuits and a latency of 100 − 200ps.

Simulation results are presented in following manner. Firstly, we show the adaptive tuning of the output-voltage swing with the resulting eye-diagrams. Secondly, we present the Q-learning results based on the I/O communication power and the BER. Finally, we compare the saving in I/O communication power as well as energy efficiency under different benchmarks of workloads.

Table 1: System settings for memory-logic integration with TSI I/O

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microprocessor</td>
<td>65nm</td>
<td>0.3mm²</td>
</tr>
<tr>
<td>Frequency</td>
<td>300MHz</td>
<td></td>
</tr>
<tr>
<td>I/O controller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dissipation power</td>
<td>5mW</td>
<td></td>
</tr>
<tr>
<td>Output-voltage swing</td>
<td>0.1V, 0.15V, 0.2V, 0.3V</td>
<td></td>
</tr>
<tr>
<td>Driving current</td>
<td>200mA, 300mA, 400mA, 500mA</td>
<td></td>
</tr>
<tr>
<td>Number of levels</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Switching time</td>
<td>10ns</td>
<td></td>
</tr>
<tr>
<td>TSI</td>
<td>3mm²</td>
<td></td>
</tr>
<tr>
<td>Inductance</td>
<td>300µH</td>
<td></td>
</tr>
<tr>
<td>Tolerance</td>
<td>±5%</td>
<td></td>
</tr>
<tr>
<td>SRAM</td>
<td>16 KB</td>
<td>0.2mm²</td>
</tr>
<tr>
<td>Power consumption</td>
<td>60mW</td>
<td></td>
</tr>
</tbody>
</table>

5.1 2.5D I/O Eye-diagram

The characteristics of eye-diagrams with the driver currents are presented in Fig. 7 by introducing 10% clock cycle-to-cycle jitter (noise) at the TSI I/O channel. Note that different driving currents can make different eye openings under the noise in channel. A larger eye opening is associated with a higher current driving ability (or a larger output-voltage swing), which can reach 190mV peak-to-peak signal swing with 300mV output-voltage swing and 95ps timing margin. Compared to the lowest level at 76mV signal swing with 100mV output-voltage swing and 77ps timing margin, one can obtain nearly twice the eye amplitude (signal swing) and the according reduction in BER at the cost of triple the output-voltage swing. Thus, one can leverage the trade-off between the power reduction and the necessary BER.

One can observe that for stage 3, without tuning the tail current, the eye opening is 96mV, but the eye opening increases to 112mV by adaptively changing the current. Similar improvement in eye openings is shown for other stages as well.

5.2 BER versus Power Curve

We further discuss the trade-off curve of the I/O communication power and BER under the change of output-voltage swing. Fig. 9 shows the change in BER (black rectangle) and the I/O communication power (blue circle). With the increase in output-voltage swing, BER decreases at the cost of I/O communication power. For example, at an output-voltage swing of 350mV, I/O communication power is 0.56mW with a BER of 4.14E − 6; whereas at an output-voltage swing of 400mV, BER goes down to 4.54E − 8 at the cost of increased communication power to 1.14mW. Since different applications can tolerate different amount of errors, I/O communication power thereby can be traded off for BER.

5.3 Adaptive I/O Control by Q-learning

To obtain an optimal trade-off between the I/O communication power and BER, we further discuss the self-adaptive tuning of the output-voltage swing based on the Q-learning Algorithm 1. The learning rate α and discount factor γ are set as 0.3 and 0.7 respectively. Auto-regression (AR) of order 8 is used for load current (or I/O communication power) prediction. Fig. 10(a) shows the I/O communication power trace and the corresponding predicted one using AR for the bzip2 benchmark. The error between the predicted and actual values are less than 0.3%. Fig. 10(b) further shows the I/O communication power trace for different output-voltage swings.
communication power by tuning the output-voltage swing. There is 19.08% saving on average for the I/O communication power by the one with the adaptive tuning when compared to the one without the adaptive tuning.

![Figure 10: (a) Power trace for bzip2 benchmark: actual and predicted; (b) Power with adaptive tuning](image)

5.4 Performance Comparison with Benchmarking

![Figure 11: I/O communication power saving by self-adaptive tuning under different benchmarks](image)

The communication power saving by the self-adaptive tuning using Q-learning algorithm is illustrated in Fig. 11. Various benchmarks such as SPEC [21] and fft are used to evaluate the communication power savings. For example in Fig. 11 for the fft benchmark, the I/O communication power with and without the self-adaptive tuning are 0.127mW and 0.199mW respectively. The power consumption is 19mW with energy efficiency of 4.75pJ/bit for the I/O without the self-adaptive tuning. On an average, 21.42% power saving and 14.47% energy-efficiency improvement can be achieved by the self-adaptive tuning.

6. CONCLUSION

In this paper, we have investigated the low-power 2.5D TSI I/O with adaptive adjustment of output-voltage swing. Based on the predicted I/O communication BER and power, the Q-learning based self-adaptive control is developed to control the output-voltage swing of 2.5 TSI I/Os for energy efficiency upon the workload characteristics. Experimental results have shown that the adaptive 2.5D TSI I/Os designed in 65nm CMOS can achieve an average of 13mW I/O power, 4GHz bandwidth and 2.35pJ/bit energy efficiency for one channel under $10^{-6}$ BER, which has 21.42% reduction of power and 14.47% improvement of energy efficiency.

7. ACKNOWLEDGMENTS

This project is sponsored by Singapore MOE TIER-2 fund MOE2010-T2-2-037 (ARC 5/11).

8. REFERENCES