CHAPTER 3

Related Work

There are several techniques proposed in literature to minimize propagation delay and/or energy consumption. We now briefly discuss some of them.

A simple technique to eliminate opposite transitions on adjacent wires is to insert a shield wire between every pair of adjacent wires [6]. Shield wires are tied statically to $V_{dd}$ or ground. The shielding (SHD) technique achieves 46% delay savings over the base case (i.e., an unencoded bus) at the cost of 63 wires for a 32-bit bus. Instead of placing a shield wire between every pair of adjacent wires, spacing between wires can be increased to reduce the coupling capacitance [6]. Even though both the shielding and spacing techniques achieve the same speedup, the spacing technique is shown to be energy efficient [6].

Crosstalk preventing coding (CPC) technique is proposed in [49] to eliminate opposite transitions on adjacent wires. Though in theory it is possible to generate 46-bit crosstalk preventing codes [49] (in the case of memoryless coding) for a 32-bit bus, it is very difficult to implement the encoder for the entire 32-bit data. In order to simplify the implementation process, one has to partition the 32-bit bus into 3-bit groups and encode each group separately. Using this partitioning mechanism, the CPC technique requires 53 wires for a 32-bit bus. The CPC technique achieves nearly 44.5% delay savings over the base case.

A bus encoding technique to avoid forbidden patterns, i.e., patterns of the form 010 and 101, from a given data item is proposed in [12]. Avoiding forbidden patterns
can result in the elimination of opposite transitions on adjacent wires. This technique requires 52 wires for a 32-bit bus and achieves nearly 46% delay savings over the base case. The codec required to avoid forbidden patterns can incur significant delay, area, and power overheads.

By combining the ideas of [12, 49], overlapping coding technique is proposed in [44]. Both the forbidden transition overlapping coding (FTOC) and forbidden pattern overlapping coding (FPOC) techniques eliminate crosstalk transitions and require 48 and 52 wires, respectively, for a 32-bit data. Though the FTOC technique requires few extra wires, it incurs significant codec delay and area overhead.

A bus encoding technique to minimize power consumption and eliminate crosstalk classes 5 and 6 is proposed in [21], which requires 55 wires to encode 32-bit data. Coupling-driven bus encoding technique [19] reduces power consumption by 30% on an average, but it may not eliminate crosstalk class 4 and above and hence it is not much advantageous from the delay perspective. Another bus encoding technique to minimize both energy and delay is proposed in [18], which can eliminate only crosstalk classes 4 and 6 (but not crosstalk class 5) so that the worst-case delay is still $C_L R_T (1 + 3\lambda)$ and it requires 55 wires to encode 32-bit data.

Odd/even bus invert technique is proposed in [53] to minimize coupling energy. With half cycle delay of one of the adjacent wires, it can eliminate opposite transitions on adjacent wires and achieve 30% power savings. Though this technique is energy efficient, from the delay perspective it is not advantageous because the half cycle delay can result in crosstalk class 4 patterns such as $- \uparrow -$ and $- \downarrow -$. As a result, the net delay for receiving the data becomes $C_L R_T (2 + 4\lambda)$, which is more than that of crosstalk class 6. Another technique to minimize power consumption due to coupling transitions is proposed in [41], but it may not be easy to implement due to its complex codec circuitry.
A bus encoding technique for system-on-chip buses to minimize power consumption and eliminate opposite transitions on adjacent wires is proposed in [17]. This technique requires 55 wires for a 32-bit bus to achieve nearly 46% delay savings along with 38% energy savings for systems implemented in 0.18\(\mu\)m CMOS technology.

In dual-rail coding technique [32] for low energy consumption fault-tolerant bus, a given data is duplicated and both the original and duplicated bits are placed adjacent. Non-uniform spacing is applied on top of the coding in such a way that the spacing between two different data bits is more than the spacing between the identical data bits. Spreading the transitions of the bits over the codeword to avoid opposite transitions on adjacent wires and the application of non-uniform spacing can result in energy savings and power-delay product reduction with respect to the Hamming codes.

A bus encoding technique using a variant of binary Fibonacci representation is proposed in [27] to eliminate crosstalk delay. The authors also gave a recursive procedure to generate crosstalk delay free Fibonacci codewords. The generated Fibonacci codewords are similar to that of memoryless self-shielding codes given in [49]. It is shown that \(m\)-bit crosstalk delay free Fibonacci codewords are used to encode \([\log_2(F_{m+2})]\)-bit bus, where \(F_{m+2}\) is the \((m + 2)\text{th}\) Fibonacci number. So, a 32-bit bus can be encoded using 46-bit Fibonacci codewords.

A technique called no adjacent transition (NAT) coding [47] is proposed to eliminate crosstalk transitions. In the NAT coding technique, by considering a restriction on the number of 1s present in the codewords, the authors have given \((n, b, t)\text{-NAT codes, where } b\text{ is the dataword width, } n\text{ is the codeword width, and } t\text{ is the maximum number of 1s allowed in the codewords. The NAT codes are transmitted using the transition signaling technique [45]. For } n\text{-bit codewords, the maximum number of } (n, b, t)\text{-NAT codes is } \sum_{i=0}^{t} C(n + 1 - i, i), \text{ where } 0 \leq t \leq \left\lfloor \frac{n+1}{2} \right\rfloor [47].
Selective shielding technique is proposed in [28] to eliminate opposite transitions on adjacent wires (crosstalk transitions). Compared to the conventional shielding technique, the selective shielding technique significantly reduces the number of extra wires. The authors gave a lower bound on the number of wires required to encode \( n \)-bit data using the selective shielding technique. For a \( n \)-bit data, this technique requires \( \frac{3n}{2} \) bits to eliminate crosstalk transitions. It is shown that the selective shielding technique achieves better energy savings and requires less area as compared to the other techniques.

Instead of using redundancy in space, a novel technique is proposed in [30] by considering redundancy in time for delay and peak power minimization for on-chip buses. The main idea of this technique is to reduce the bus width and transmit data in multiple cycles. The reduction in the bus width is exploited to increase the spacing between wires. By exploiting the fact that the coupling capacitance decreases with increasing spacing between wires, this temporal redundancy based technique achieves better delay savings.

Unlike the above techniques which consider fixed clock period for data transmission, by exploiting data transition behavior of SPEC2000 CPU benchmarks, variable cycle transmission (VCT) technique [20] is proposed in which different data items are transmitted with different delays and the necessary delay for a data item to be transmitted is determined based on the crosstalk class of the data item with respect to the present data on the bus. This technique requires 34 wires for a 32-bit bus. As an extension to this technique, a bus encoding technique for delay and energy efficient data transmission for on-chip buses is proposed in [29]. In this technique, variable delay for data transmission [20] is exploited to apply temporal redundancy in the process of minimizing both delay and energy for on-chip data transmission. It requires 35 wires for a 32-bit bus.