CHAPTER 2 RELATED WORK

The previous work has been proposed for solving crosstalk problems on buses, reducing power dissipation of fetching and decoding instructions, and low power multiplier designs. In this chapter, we survey and discuss the related work.

2.1 Related work on minimization of crosstalk

In the past, two cost functions, the minimization of total crosstalk and the minimization of maximum crosstalk, have been proposed to solve the crosstalk problems on buses. The main objective of minimizing total crosstalk is to reduce power induced by the coupling capacitances. Several methods have been proposed to minimize the total crosstalk. In [1][9][10], several methods were proposed to solve crosstalk problem by changing wire space and permuting wire sequence. In [11][12], codec techniques by changing signal phase were proposed to minimize crosstalk on bus transitions.

The main objective of minimizing the maximum crosstalk is to guarantee performance. Several methods have been proposed to minimize the maximum crosstalk. In [13], a method was proposed to generate “self-shielding codes” on buses for avoiding crosstalk sequences. In [14][15], the authors classified crosstalk interactions between bit-lines on a bus and presented several encoding techniques that can support designers to trade off crosstalk against the area overhead. In [16], a codec technique for eliminating crosstalk sequences and lowering power was proposed. These methods can achieve significant crosstalk reductions with paying 25-43% area overhead. By utilizing the knowledge of address bus transition sequences, the authors

of [17] proposed a bus encoding technique to save the wire overhead. In this thesis, we will address the crosstalk issue on minimizing the maximum crosstalk.

2.2 Related work on low power instruction decoder designs

In the past several years, many techniques have been proposed to reduce power dissipation of fetching instructions. The proposed techniques are classified into three categories.

The first group reduces instruction-fetching power dissipation by applying program-compression and bus-encoding techniques. In [18][19], several techniques were proposed to compress the most commonly executed instructions for energy-dissipation reduction in memory accesses and to decompress the instructions before the decoding stage. In [20], a technique was proposed to reschedule program codes to minimize the transitions on the instruction bus. In [21], a bus-invert encoding technique was proposed to reduce power on the instruction bus transmission.

The second group of techniques uses additional instruction buffers to store fetched instructions in processors. In [22], a "*decoded instruction buffer*" was introduced to turn off fetch-units when instructions in a loop are executed. Two methods "*Decode filter cache*" [23] and "*Micro-operation cache*" [24] were proposed to save instruction-fetching power by storing the executed instructions.

Finally, the last group proposes new instruction-set architectures for low power designs. The MIPS architecture [25] and the ARM Thumb architecture [26] use new instruction formats with reduced instruction lengths so that memory traffic and hence power can be reduced.

Most previous studies focused on the power reduction of fetching instructions.

Only a few studies focused on power reduction for instruction decoding and control signal generation. A two sub-decoders structure [27] was proposed for the power minimization on instruction decoders. In [28], two types of instruction decoders were designed for Pentium Pro: one for simple instructions and the other for complex instructions. This structure was proposed for performance improvement rather than power minimization. The studies have also shown that the instruction decoder can consume as much as 18% of the system power in *StrongArm* [4] and 14% in Pentium Pro [24]. Thus, this thesis focuses on the reduction of instruction decoder power usage.

2.3 Related work on low power multiplier designs

Power consumption has become one of the most important design issues for DSP designs targeted to multimedia and handheld applications. An important trend for low power DSP designs is to customize the instruction set for accommodating program characteristics with ASIP designs (Application Specific Instruction-set Processors) [5][29]. In [30][31][32], low power instruction-set techniques for ASIPs have been proposed. For low power memory access and data-path, several techniques, including instruction buffering [33], bit-width optimization [34][35], and clock-gating [36], were also proposed.

Due to the large area, critical timing and high power dissipation, multipliers are the most critical components in an application-specific design. Two approaches [8][37] were presented for low power multiplier designs. In [37], an ensemble of point structure was presented for low power multiplier designs. In [8], a configurable structure modified from ensemble of point structure was proposed to reduce the area

overhead. The key idea of a configurable multiplier-structure [8][37] to save power consumption is that the multiplier has two configurations. When the smaller multiplication is performed, unused parts of the multiplier are turned off. This technique has been proven to be very effective in power reduction. Nevertheless, these techniques focus on ASIC (Application Specific Integrate Circuit) designs rather than processor designs. Moreover, the bit-width of the smaller configuration is simply chosen to be the half of the bit-width of the larger multiplier.

However, we have observed that a smaller configuration with half of the maximum bit-width is not necessarily a good choice for the bit-widths distribution of multiplication instructions for some specific applications. We will present a technique to utilize the architecture [8] and determine the bit-width of multiplication instructions.