Area – Time – Power and Design effort: the basic tradeoffs in Application Specific Systems

Michael J Flynn
Stanford University
flynn@ee.stanford.edu

Abstract

Application specific is always a tradeoff among competing design goals (or design parameters). In addition to the well established area (cost) – time (performance) –power metrics specific applications imply a relatively limited market so design cost becomes an especially important consideration. As technology offers increasing transistor density with lower cost power constraints limit frequency as the primary avenue to performance. The alternative is to use area (transistors) to recover performance putting an additional strain on the design budget. The search for flexibility in design without paying a significant Area – Time – Power cost remains the primary problem for application specific and System on a Chip (SOC) design.

1. Introduction

The current ITRS technology [1] roadmap promises 100x circuit density and 10 x clock speeds in the next fifteen years. As in the past, this roadmap is more accurate in predicting circuit density than clock speed because it is much harder to foretell the effects of variations in future circuits and microarchitecture. Nevertheless, it is very unlikely that clock speed will exceed *10 x* due to significant power limits. Assuming we can fully utilize the extra area without any major communication or other performance overheads, this represents up to *1,000x* processor performance or (as we will see) $10^{-6}x$ power consumption. How these enabling technologies are used depends on the target application and the amount of design effort needed to realize the advantages.

2. Traditional AT (Area-Time) tradeoffs

Traditionally microprocessor and microarchitecture design optimization is largely an exercise in studies of cost- performance or area- time tradeoff. For silicon technology this type of "textbook" design has been nicely formalized by theorists as "AT" bounds. Simply stated, if a design has more area, A, available it should be able to perform a given computation is less time, T. Ullman [2] shows that an implementation is theoretically bounded by AT if it is limited by volume (I/O, number of devices, etc.) or AT^2 in implementations that are communications (internal bisection) limited. So depending on the type of functional unit and circuit implementation, doubling processing speed may increase die size by *2* to *4* times.

$$A \cdot T^n = \text{constant (n is typically between 1 and 2)}$$

So tradeoffs are possible along this bound. Larger designs should go faster. Designs whose AT product is higher than the state of the art are inferior designs

3. T P (Time-Power) tradeoffs [3]

The situation with power and time tradeoffs depends on the type of design. For workstations new designs are increasingly limited by static power while SOC and similar designs remain limited by dynamic lower.
At the device level, total power dissipation (P_{total}) has two major sources: dynamic or switching power and static power caused by leakage current:

$$P_{total} = (C\ V^2\ freq)/2 + I_{leakage}\ V$$

where C is the device capacitance, V is the supply voltage, freq is the device switching frequency, $I_{leakage}$

Proceedings of the16th International Conference on Application-Specific Systems, Architecture and Processors (ASAP'05)
1063-6862/05 $20.00 © 2005 IEEE

is the leakage current. Until recently switching loss was the dominant factor in dissipation, but now static power is increasing. As feature sizes decrease so do device sizes. Indeed, this enables increasing performance. Smaller device size reduces capacitance correspondingly. Since,

$$I/C = dV/dt$$

with the same drain or charging current, I, the reduction in C enables faster signal transition times (higher frequency). But as device size shrink the electric field applied to them become destructively large. One volt difference across a 0.1 micron insulator is the same as 100,000 volts across one cm. So if we avail of the density and higher frequency, we have to reduce the supply voltage. In turn for correct circuit operation we have to reduce the device, V_{th}, threshold voltage (the point of logic level switch). Bringing V_{th} closer to ground (the off level) increases the leakage current. This has an important effect on design and production. There are two types of device designs that must be accommodated in production:

1. The high speed device with low V_{th} and high static power.
2. The slower device maintaining V_{th} and V at the expense of circuit density and low static power.

In either case we can reduce switching loss by lowering the supply voltage, V. Chen et al.[4] showed that the drain current is proportional to

$$I = (V - V_{th})^{1.25}$$

where, again V is the supply voltage. From our discussion above, the signal transition time and frequency scale as the charging current. So the maximum operating frequency is also proportional to $(V - V_{th}^{1.25})/V$.
For values of V and V_{th} of interest this means that frequency scales as the supply voltage, V.

Assume V_{th} is 0.6V, suppose we reduce the supply voltage by 1/2, say from 3V to 1.5V, the operating frequency is also reduced by about 1/2. So reducing the supply voltage by half also reduces the operating frequency by half. Now by the power equation (since the voltage and frequency were halved) the total power consumption is 1/8 of the original. Thus, if we take an existing design optimized for frequency and modify that design to operate at a lower voltage, the frequency is reduced by approximately the cube root of the original (dynamic) power, or

$$T^3 P = \text{constant}$$

As power becomes increasingly important the cubic tradeoff between time and power ($T^3P=k$) forces designers to use relatively cheap area to increase performance rather than expensive power required by higher clock rates. As designers focus attention on power the question of optimal power oriented architectures is apparent

4. A T P design optimization

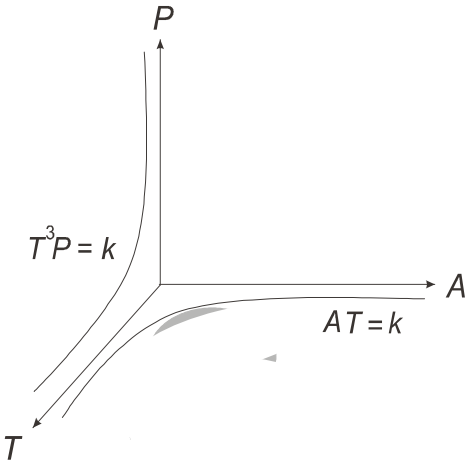

Figure 1. Design space outline of area – time - power

Putting these bounds together defines a surface of optimal A T P design possibilities, as shown in figure 1. Designs that are interior to this surface are non-optimal designs… at least from the A T P point of view. The problem now is that it may take excessive design effort to achieve such an optimal design.

5.0 Design effort ($\mathcal{E}$)

Over the past 20 years, die complexity (transistors per die) has grown at 58% per year, but designer productivity (CAD tools, etc) has only increased by 21% per year [5]. This design gap (Figure 2) is formidable and has served as the basis for many design ideas. Figure 2 shows a productivity gap increasing at the rate of about 25% per year. So, unless the production run of a design is quite large, the A T P

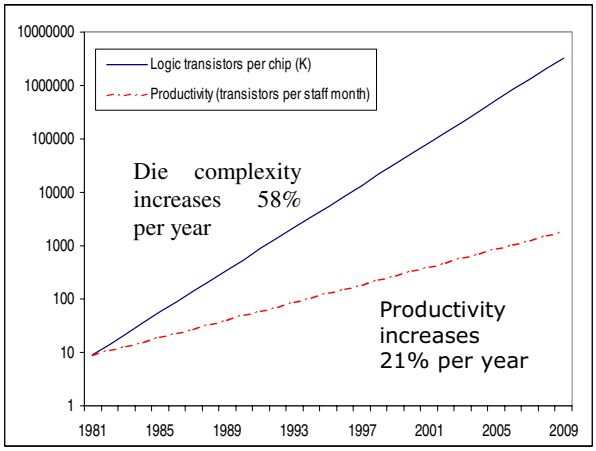

Figure 2. Design complexity (transistors per chip) and design productivity as a function of time.

optimization must be secondary to realizing good (A T P) designs at a reasonable design effort ($\mathcal{E}$). Indeed, at the very moment that large transistor budgets become available for integrating complete application specific systems, the design budget constrains users to make more modest technology choices. How ambitious depends on the market size over which the design cost can be written. At one extreme systems can be realized with FPGAs which can be reconfigured to suit any particular application. At the other extreme is the completely custom design. The tradeoff is between design time ($\mathcal{E}$) and the flexibility (or re use or programmability) of the design. This flexibility is achieved at the expense of the A T P product. So there is also some sort of another bound:

Flexibility * design effort = constant

More flexible designs require less design effort overall applications, since they are applicable to more design applications.

It is in the area of flexibility (or design re use or programmability) that the field faces the greatest challenge.

6.0 Acknowledgements

This paper is based on joint work with Patrick Hung. A more complete version is currently about to be published [6].

7. 0 References

[1] International Technology Roadmap for Semiconductors 2003 Edition. (http://public.itrs.net/).
[2] J. D. Ullman, "*Computational Aspects of VLSI*." Computer Science Press, 1984.
[3] M. J. Flynn, P. Hung and K.W. Rudd, "*Deep-Submicron Microprocessor Design Issues*." IEEE Micro Magazine, July-August Issue, 1999, pp. 11-22.
[4] K. Chen et al., "*Predicting CMOS Speed with Gate Oxide and Voltage Scaling and Interconnect Loading Effects*," IEEE Trans. Electron Devices, Vol. 44, No. 11, Nov. 1997, pp. 1,951-1,957.
[5] S. Malik, International SEMATECH (http://www.sematech.org/).
[6] M. J. Flynn and P. Hung, "Microprocessor *Design Issue*: some thoughts on the road ahead" IEEE Micro to be published 2005