Microelectronics System Design Research Group
School of Electrical & Electronic Engineering
Merz Court
Newcastle University
Newcastle upon Tyne, NE1 7RU, UK

http://async.org.uk/

# On balancing regularity and burstiness in architecting new computer systems

Alex Yakovlev

alex.yakovlev@ncl.ac.uk

NCL-EECE-MSD-MEMO-2012-005

February 2012

Biological systems typically have two types of operation, regular and bursty, and manage to organise their operation in an energy efficient and robust way, which also supports natural tendency for survival. Regular activities take place all the time, and are meant to serve the needs of the overall system and are determined by the overall structure and dynamics of the system. Bursty activities are typically not those that are constantly triggered by normal periodic cycles of the system, but rather they are triggered by or in accordance with the needs to react to the demands of the environment. Why not to build a computer system in a similar fashion, such that a constantly active part has to be relatively slow and all the fast processing has to be done in specialised units, whose activation is bursty?

Biological systems typically have two types of operation, *regular* and *bursty*, and manage to organise their operation in an energy efficient and robust way, which also supports natural tendency for survival.

*Regular activities* take place all the time, and are meant to serve the needs of the overall system and are determined by the overall structure and dynamics of the system. Intuitively, for small size systems regular activities are fast and for large systems they are slow. Regular activities are typically synchronized well to the supply and distribution of energy inside the system, the process which is itself regulated in an energy-efficient way. Indeed, the power supply is itself an activity which is controlled by some sort of a central managerial core of the system. In other words, something that is supposed to work all the time is better to be pulsed regularly rather than irregularly, and with minimum energy losses during each pulse. The rate of change during each cycle is typically associated with some form of charging and discharging of energy buffers, and it is known that if the rate of change is moderately slow (so called adiabatic cycling), the amount of heat dissipation during charge-discharge cycles is minimal.

*Bursty activities* are typically not those that are constantly triggered by normal periodic cycles of the system, but rather they are triggered by or in accordance with the needs to react to the demands of the environment. Such activities are dealt with by special-purpose subsystems, whose energy resources are supplied by the action of the system's core, and whose dynamics of the operation is determined by their local structure and dynamics. Thanks to the biological evolution, the optimality of response to bursts is determined by the best adherence to the duty cycling needed to respond to such bursts. Of course, the ultimate level of effectiveness and efficiency in bursty modes will be determined according to the amounts of energy allotted to these subsystems by the core, which is the mains supply of energy. Intuitively, subsystems that operate in bursty way are typically locally timed (according to the limit cycles of the subsystem, in other words according to the properties inherent in its topology). The rate of change of charging and discharging of the energy buffers of subsystems can be relatively high because this is how they reach optimum energy utilisation in those bursts of activity. For example, a burst must be dealt with with minimum energy leakage as well as with the requirement to produce an effective result within the allotted interval of power supply. Typically, subsystems that deal with bursty activities are peripheral to the core subsystem.

We can postulate, again building on the analogy with biological systems, that the duty cycling of bursty activities is completely different from that of regular activities. Since bursts are typically served with much higher rate, they are relatively irregular and happen relatively infrequently. Thus, if we look at the system from the point of view of the average wear and tear, we can see some form of the overall balance of the integral amount of activity of organs in the core and peripheral parts, which intuitively helps the overall system to age gracefully and uniformly.

Now let's move to the world of computer systems, and try to treat them as systems which we would like to lead harmonious and balanced life, and perform their activities in such a way that they can both (i) serve their own needs of staying alive and functioning efficiently (regular action), as well as (ii) serve the needs of their user and/or environment which drives them with specific tasks (bursty action). How to categorise activities between regular or bursty? Are there already examples of such sub-division in computer and electronic systems?

We can have a look at the functionality of a simple microprocessor as our first example. It is fairly obvious that the main instruction processing control has to be active all the time if its main purpose to maintain life in all parts of the system at all times. But does it have to be very fast? Well, it has to be fast if the instruction control is driven by the needs of the environment. But if our assumption is that only bursty processing can be fast, then the environment must be bursty, and hence instruction processing is not regular. In our approach, where we have to be realistic about resource efficiency and robustness, we will postulate that the full system cannot have both regular and high rate!

With this postulate, we will have to organise the processor in such a way, that a constantly active part has to be relatively slow and all the fast processing has to be done in specialised units, whose activation is bursty. In this processor the main control cycle can be driven by the power supply which resembles "adiabatic", with slow charging and discharging, and its responsibility is to dispatch requests for data processing jobs to the peripheral units. The job requests are sent out together with the allocation of access to the appropriate power line at the time when this line has a certain level of Vdd, during which the unit's own clock generation subsystem will trigger the computational activity in the most power-performance efficient way. For example, the local clocking can be done through the local oscillator whose timing is scaled well in agreement with scaling the worst-case cycle of the logic of the local unit (such things are often done by using a replica of the worst case cycle in the clock generator). Alternatively, the local unit can be built using speed-independent logic with completion detection and its operation in time is modulated by the current value of the Vdd supplied to the whole unit. All such power modulated modes of operation are guaranteed by the appropriate level of elasticity of the logic implementation of the unit.

As our second example, let's consider a many-core multiprocessor. In such a system, the role of the regular activity organ goes most likely to the power distribution and communication infrastructure, which must always be enabled and ready for action, while the cores can be on and off depending on the needs coming from the application tasks. On the other hand, making the interconnect fabric slow is not always possible, and in fact where we have to strive for throughput we have to enable it for fast action. So, how can we resolve this conflict between regularity and high speed? Well, one cannot have the cake and eat it. We have to make some sacrifices. And for that we need to put quite a significant amount of redundancy in the interconnect, either to provide 'cold spare', which can be substituted when the hot parts fail, or have some form of dynamic reconfiguration in which the artificial (internal or system-driven) 'quasi-burstiness' is introduced, so that fast organs are given time to rest. This could for example be done by certain time or space division demultiplexing.

A schematic representation of the idea of an activity balanced system with a regularly pulsing core and bursty periphery is shown in Figure 1. It shows a fairly general case in which there are two types of power supply, one standard with stable Vdd level and the other adiabatic. This allows support of the execution of tasks that fall in various classes.

Let's first consider the class of regular tasks. These tasks are typically 'reference tasks', which are highly periodic compared to bursty or sporadic tasks. Reference tasks determine the 'functional face' of the system as a whole in terms of performance, energy and reliability. They are setting the dynamics of the system according to its main structural properties. Depending on the level of granularity of periodicity we can divide regular tasks further into two categories. The first one, which we call here "high frequency regular tasks" are supposed to be executed at the guaranteed level of performance and timing, and hence are driven by clock while power level is always maintained stable (this of course does not prevent us from allowing mechanisms such as voltage scaling to be applied in this class to switch between different modes of the system-level execution). The second one, which we call "low frequency regular tasks" are executed in an adiabatic mode, hence their timing is governed by the dynamics of the system's power supply.

In the class of bursty tasks we can place tasks that are more sporadic and those that are triggered on demand,

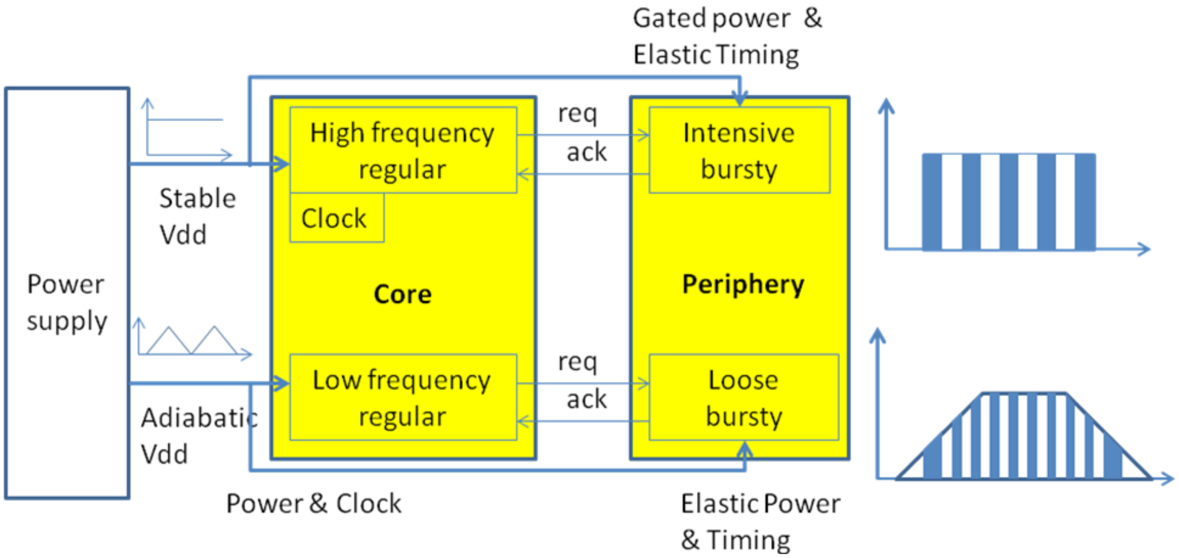

Figure 1: System architecture

as determined either by the needs to respond to the environment stimuli or according to particular types of computation functions triggered by the commands from the core part (e.g. executing instructions). Again, here we can have two categories. In the first, which we call "intensive bursty" we have tasks which are regulated by the level of supplied voltage. This level, although it modulates the rate of the task execution, by being stable allows the task to serve the needs of computations with well defined timing (e.g. sampling rate). So, here we have timing elasticity due to different voltage levels and request-acknowledgement interaction with the core part. In the second category, called "loose bursty" we allow maximum elasticity, because the power level coming to them is not constant and is varied according to the waveform from its adiabatic source. In combination with the "on demand" request-acknowledgement interaction with the core, the timing of each cycle of computation is determined by the instantaneous value of the power signal. These tasks are somehow elastic both in time and voltage levels.

Let's now step back and see why the above two dimensional classification is useful.

In order to do that we need to relate the nature of applications and functionalities of the system with the approach to modelling and analysis of power and timing regimes that could support those functionalities. Suppose we have a description of the system's main functionalities. The power and timing regimes to serve these functionalities can be worked out by analysing the shapes of their duty cycling. This is analogous to having some sort of "spectral analysis" of their activation. The duty cycling is clearly divided between regular and sporadic on one axis, and between time-driven (intensive) and power-driven (loose) on the other axis. On the other hand, from the timing point of view we have two levels of granularity. The coarse level is the level at which the task is started and finished, and the other level is the fine granularity level in which the task is cycled between its micro-steps. At both levels we can consider a range of degrees of timing elasticity, which are related to the profile of the power supply voltage dynamics, hence our subdivision between "timing elastic" and "timing and power elastic".

It is probably unlikely to expect that all of the above forms of timing and power regimes will be present in one particular processor design. However, systems become more diverse and their qualities are measured on a multi-parameter scale, including the environmental conditions such as resource supply and levels of uncertainty, and the features such as efficiency and robustness. Thus, setting the goal of designing a system of the kind of the above at the architectural or even logic level appears to be timely and novel.

Explanatory note on adiabatic switching and power-clocking: For CMOS electronic circuits the adiabatic supply mode involves slow ramps of rising and falling level of Vdd (smooth operation). This way the amount of dissipated energy can be reduced by the factor of $\frac{2 \cdot T}{R \cdot C}$ on the cumulative load capacitor $C$ through the parasitic resistance $R$ of the transistors that are switched on; here the time of the ramp is $T$. If $T \gg RC$, the energy savings are significant, and could allow the system to live longer, but of course at the cost of slower operation. The requirement of the slow operation of such core elements is fundamental because these elements perform their computational actions according to the rate of the power supply, i.e. their data changes are synchronous to the power transitions. Such operation is often called power-clocking or clocking by power signal.