# Pareto analysis

The Pareto analysis technique is derived from economics analysis and is used for selection of a limited number of tasks that produces significant overall effect. In short it’s about gathering the task which produces the most and cost the less. An pareto optimal solution would occur when no further pareto improvement can be made.

The technique is assimilated to the 0-1 Knapsack problem, and can be used to compare different hardware architectures. Normally pareto analysis is based upon cost and performance. While these factors are not the only important parameter in a project, we would like to give a proposal which include performance and risk as an alternative. A pareto analyse of available components for emergency call button is illustrated below.

|  |  |  |
| --- | --- | --- |
| **Pareto analyse Emergency call button** | | |
| **Architecture** | **Execution Time** | **Cost** |
| 1. Microcontroller and ISM transceiver ASIC (CC430) | 0,1516 | 18,15 |
| 2. DSP, Microcontroller and ISM transceiver ASIC. (TMS320VC5401) | 0,05264 | 23,57 |
| 3. Microcontroller and ISM transceiver ASIC (PIC16F1516) | 0,6064 | 10,97 |
| 4. FPGA, ADC/DAC and ISM oscillator and LNA receiver filter. | N/A | 1295 |

Tabel 1

The architectures are the ones described in chapter 7.2. Executions times are derived from examine the executions paths described in chapter 7.2 and thereafter calculated the estimated load. Loaded can be calculated as:

*Load(PE) = fs \* Sum (ops/sample) / fpe*

|  |
| --- |
|  |

Figur 2

All design point for Emergancy call system is illustrated in the figure above. Design point consist of an allocation (MIPS, DSP , FPGA ), and bindings to assignment. Each assignment can be bounded to different allocation. Because our product deals with sampling rates of 8kHz, and some relative simple encode/decode algortimhes, only the audi block need calculation power, so the mapping of assignment to allocation is already given and described in chapter 9.1 (Mapping of the general architecture).

The graph in figure 17 illustrates some of the design points described in chapter 9.1. Only the FPGA is missing, which is because we don’t really know the execution time. The algoritmens in the FPGA runs in parallel which means we have run examples implemented in HDL to examine executions time, which is beyond this project, in contrast to all other pareto design point which is implemented in a sequential processor.

Also the graps shows a design point which is not described in the mapping chapter 9.1 and that’s the PIC16. The PIC16 is the cheapest MIPS and runs only at 5Mhz, and a low power consumption of only 600nA.

The interesting thing about the PIC16 design point is that the total slack in very low, compared to the other design point which have a huge slack. Slack is free cpu time and is calculated as:

Slack(PE) = Capacity(PE) – Sum Ops(p) \* fs

System capacity is 1, to process data in real time. Above 1 means we can’t handle data real time.

Slack(PIC16) = 1 - 0,6064 = **0,3937**

The PIC16 has the same cost price as the lightning fast DSP design point, but uses far less power. If the requirement to the system doesn’t change must in time, the PIC16 could a possible solution.

When we analyze the graph in figure 17 we can see that, there is no pareto optimal point, because no point is generally better in cost or performance then another design point. If we had a pareto optimal point, this point could be a god candidate for platform is the risk also were low. But as this is not the case we would have to choose a platform based on other properties, such as cost, power and risk.

Pareto analysis doesn’t tell anything about risk. Because risk of technical failure is indeed necessary to take into account, we can exclude the FPGA solution as a possible platform. The risk analyze in chapter 7 gave the FPGA platform the highest rank possible, and that is also one of the reasons that is not a design point in the pareto analyze.

If the pareto analysis had another dimension say risk(z), (x, y, z) it would be more useful, and give system engineers better discernment from the given information.

The remaining design points in the graph are all capable of doing the job, and they all have different pro and const. The CC430 has the lowest rank on risk, and a god price, and seems as a straight forward choice. On the other side the PIC16 could do the job with the current requirement. The DSP solution is pricy and incredible fast, and could meet future requirement easy, but the technical risk is a bit higher.