Design and Simulation of a pipeline-structured Floating Point Unit for high performance general purpose processors

Eduardo Balliriain, Martín I. Falcón Faya, Pablo Slavkin, Norberto M. Lerendegui*

Instituto Tecnológico de Buenos Aires, Departamento de Electrónica,
Av. E. Madero 399, (C1106ACD) Buenos Aires, Argentina
* nlerende@itba.edu.ar

Abstract

This paper presents the design and simulation results of a floating point unit that complies IEEE 754 standard for single precision operations. The architecture of the proposed FPU, that has been named PFPU, uses a pipelined core capable of enhancing its operations flow up to 600 %. The simulation results show that operation processing speed presents at least a 50% improvement compared to top-of-the-line commercial systems, such as Pentium IV.

The versatility of the proposed PFPU makes it perfectly suitable for a great range of applications, like common PCs, Digital Signal Processors and portable equipment. In addition to this, a power-save feature that allows a power dissipation reduction ranging from 42% to 82% is also presented.

1. Introduction

In both scientific investigations and engineering it is very common to use the well known scientific notation, which represents a real number as a mantissa (fraction) and an exponent, that is associated to a base-10 power. This numerical representation system covers a huge range using very few digits, mostly because all physical magnitudes can be measured with limited precision.

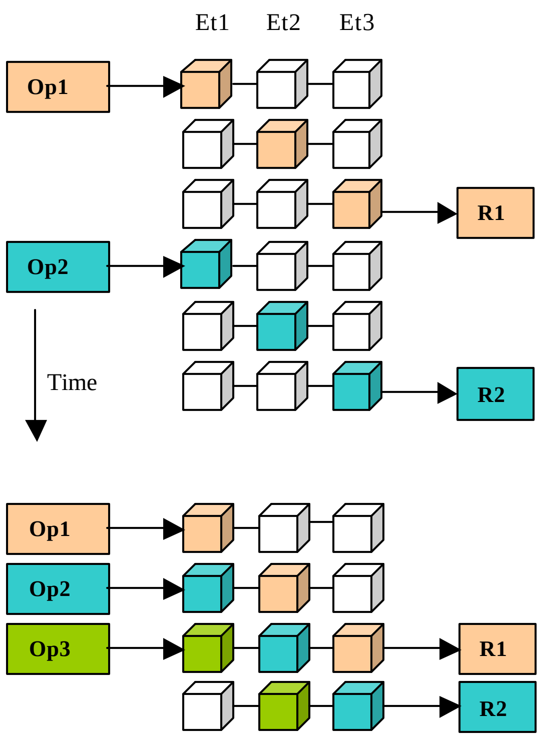
First digital processors were developed with a very limited mathematical capacity, based mainly in the addition of single and double precision numbers. With the evolution of technology, enhancements in Arithmetic-Logical Units (ALUs) leaded to the implementation of multiplication and division instructions for integer numbers as well as multiple precision operations. Floating point operations became common with the arrival of the personal computer concept, and were successfully implemented in a hardware module separated from the main processor, called coprocessor (e.g. Intel 8087), which worked on request. Due to both floating point operations requirement and IC integration technology, the coprocessor became an internal module in all PC processors, called Floating Point Units (FPU) [1].

Independently of the particular FPU architecture, it became necessary to establish a binary coding system that could offer an important dynamic range while spending few bits. This system should also be compatible with the standard processor data bus size, generally 32 or 64 bits, and provide an industry standard to allow different companies to produce interchangeable hardware. This important issues were addressed by a

Berkeley professor named William Kahan, whose work led to the IEEE 754 standard ([2], [3], [4]).

IEEE 754 standardizes two floating point numerical formats: single precision (32 bits) and double precision (64 bits). It also states that a number is organized in three sections: Sign, Mantissa (or Fraction, which is always positive) and exponent, involving 1, 8 and 23 bits, respectively, in single precision, and 1, 11 and 52 bits in double precision. The formal expression is: $N = S*M*.2^{EXP}$. The exponent is used in 'Excess 128' mode for single precision, and 'Excess 1023' mode for double precision. This means that 128 or 1023 has to be subtracted from the number stored in the exponent section in order to get the actual exponent number. This implementation leads to simpler and faster adders, since it makes the hardware sign-immune.

One of the greatest advances in processor architecture was the incorporation of pipelining inside the stages ([5], [6]). This concept is shown in the Figure 1, that presents the case of 3 operations carried out by a system composed of 3 serial stages. Pipelining technique involves identifying serial stages in the execution of an instruction and reordering the hardware to get all stages busy during processing.



Without Pipeline:
One operation can not be started until the preceding operation is completely finished. Therefore, most of hardware stays useless all the time (white cubes).
In the example shown, running 2 operations involves 6 time units.

Figure 1: Pipeline

With Pipeline:
It is possible to start one operation per time unit, keeping all the hardware modules running simultaneously. In the same example, the results are available in 4 time units. When many operations are in progress, all the modules will be always working, and the Operation Processing will be three times larger.

In this work the design and simulation results for a seven-stage Pipelined Floating Point Unit (PFPU), that complies the IEEE 754 standard for single precision operations, is presented. Special attention to stage number selection is devoted.

2. PFPU Design

Selection of the Optimum Pipeline Stage Number

The Figure 2 shows a simplified schema of the PFPU modules subject to pipelining.

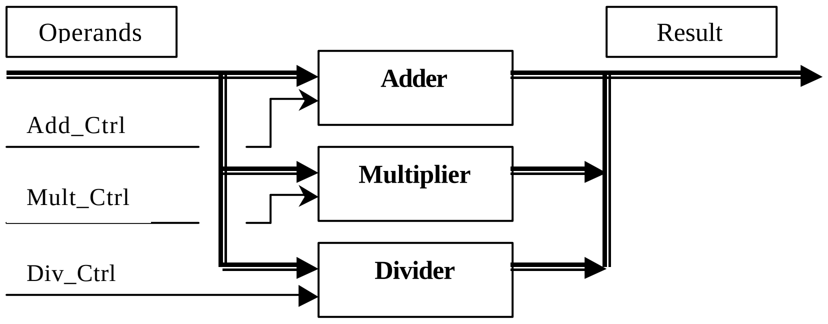


Figure 2: Simplified Block Diagram of the PFPU

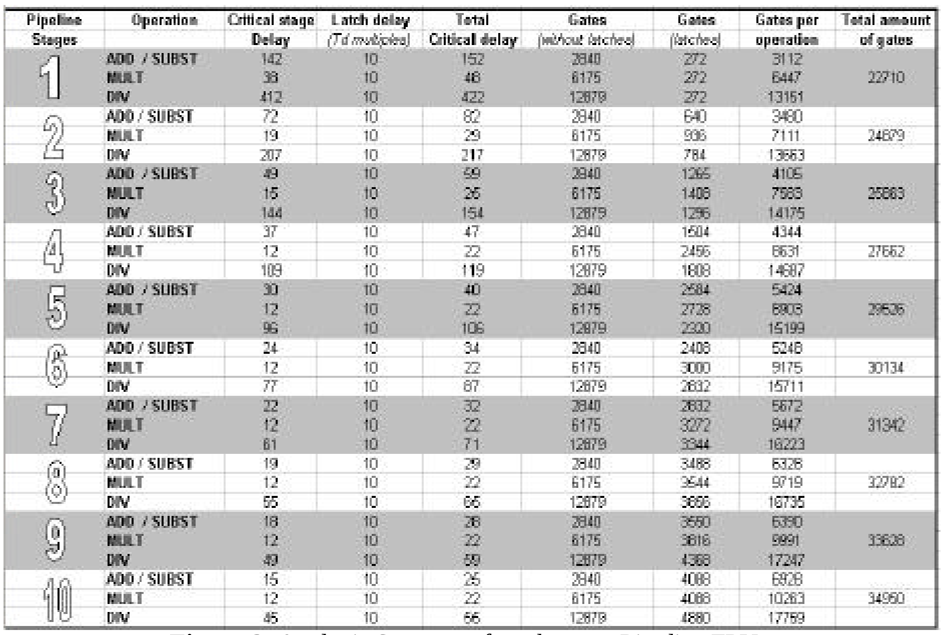
The purpose of this design was to maximize the processing speed of the PFPU. The linear relationship between speed and number of stages introduced in the previous section would erroneously suggest that an infinite number of stages give infinite speed. However two important factors should be taken into consideration:

1) Each time a stage is added to the pipeline, memory cells (latches) have to be added to the hardware so that the results of one stage can be stored until the next stage has finished processing the preceding instruction. This memory cells take some time to store the results, degrading the relationship between speed and number of stages.
2) Depending on the place selected to split up the block during pipeline stage insertion, several data lines will have to be stored, thus giving a strong variation in the associated hardware cost.

The following analysis determines optimum stage number for this design. The system performance was studied by analyzing pipelines with stage number ranging from 1 to 10.

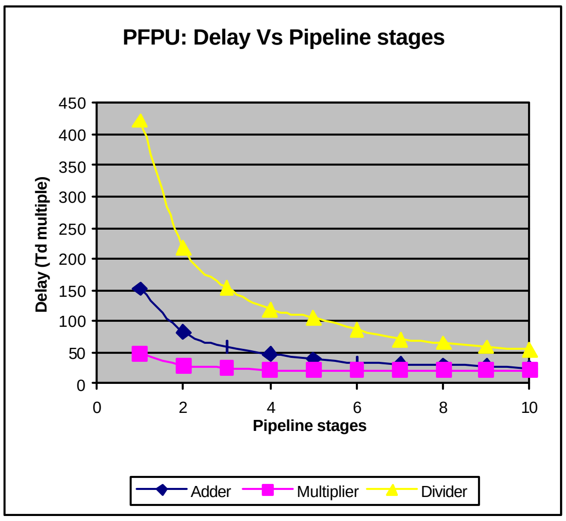
Information was gathered to identify the possible places to make the ´cuts´ to introduce the stage connection latches. As these ´cuts´ would leave asymmetric stages (e.g. it is not a good idea to insert latches inside a full adder stage, since too much hardware would be involved), the time taken by each stage differs from the others. Therefore the issue was addressed by identifying the ´critical path´, that is defined as the longest time taken for any stage to fulfil its task. This time, being the worst case, is then assigned to all stages equally (pipeline equalization). Time measurement was performed using transistor delay times (Td) as normalized units, in order to compare the design with any commercial system.

On the other hand, the quantity of gates required for each of the ten possibilities was estimated in order to get an idea of the ´hardware cost´ involved in each case. The memory cells used in this design have a time response delay of 10 Td. All collected data was normalized (forced to be between 0 and 1) so it could be used in an optimization formula (defined below). All data is summarized in the table of the Figure 3. From these table it is possible to obtain the Pipeline Delay as a function of the Number of Stages, and the Number of Gates as a function of the Number of Stages. These functions are plotted in Figures 4 and 5 , respectively.



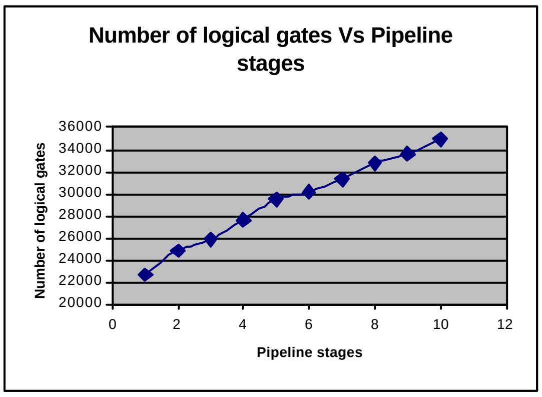
Pipeline Stages	Operation	Critical stage Delay	Latch delay *(Td multiples)*	Total Critical delay	Gates *(without latches)*	Gates *(latches)*	Gates per operation	Total amount of gates
1	ADD / SUBST	142	10	152	2840	272	3112	
	MULT	38	10	48	6175	272	6447	22710
	DIV	412	10	422	12879	272	13151	
2	ADD / SUBST	72	10	82	2840	640	3480	
	MULT	19	10	29	6175	936	7111	24679
	DIV	207	10	217	12879	784	13663	
3	ADD / SUBST	49	10	59	2840	1265	4105	
	MULT	15	10	25	6175	1408	7583	25863
	DIV	144	10	154	12879	1296	14175	
4	ADD / SUBST	37	10	47	2840	1504	4344	
	MULT	12	10	22	6175	2456	8631	27662
	DIV	109	10	119	12879	1808	14687	
5	ADD / SUBST	30	10	40	2840	2584	5424	
	MULT	12	10	22	6175	2728	8903	29526
	DIV	96	10	106	12879	2320	15199	
6	ADD / SUBST	24	10	34	2840	2408	5248	
	MULT	12	10	22	6175	3000	9175	30134
	DIV	77	10	87	12879	2832	15711	
7	ADD / SUBST	22	10	32	2840	2832	5672	
	MULT	12	10	22	6175	3272	9447	31342
	DIV	61	10	71	12879	3344	16223	
8	ADD / SUBST	19	10	29	2840	3488	6328	
	MULT	12	10	22	6175	3544	9719	32782
	DIV	55	10	65	12879	3856	16735	
9	ADD / SUBST	18	10	28	2840	3550	6390	
	MULT	12	10	22	6175	3816	9991	33628
	DIV	49	10	59	12879	4368	17247	
10	ADD / SUBST	15	10	25	2840	4088	6928	
	MULT	12	10	22	6175	4088	10263	34950
	DIV	45	10	55	12879	4880	17759	

Figure 3: Analysis Summary for a k-stage Pipeline FPU



1) Independently of the number of stages taken, the time used for all stages is completely defined by the divider circuit, since this circuit is much slower than the multiplier and the adder. Accordingly, both the adder and multiplier will have idle time.
2) The stage delay decreases with the number of stages in a **non-linear** way. This is due to the fact that the shorter the stages (by increasing its number) the stronger the effect of the constant delay of the latches (10 Td per latch). As shown in the table, for 10 stages 22 % of the delay is caused by the latches.

Figure 4: Pipeline Delay as a function of the Number of Stages



3. The number of logical gates used grows linearly with the number of pipeline stages to implement. The reason relies on the fact that the cutting places of the circuit were chosen carefully. This conclusion suggest that the circuit is scalable.

Figure 5: Number of Gates as a function of the Number of Stages

The data in the Table of the Figure 3 indicates that the bottleneck of the system is the Divider block, responsible for the longer delays. Since there are several engineering applications in which this block is not required (e.g.: DSP: Digital Signal Processing), a special operation mode was envisaged for the design. In this mode, named DSP, there is no divider block and the critical path is related to the Adder block (as it can be seen in the table).

A decision criterion was chosen to get a good trade-off between speed and hardware cost: the smaller the number of gates and the shorter the delay, the better the design. The graphs presented in the Figures 4 and 5 show that the Pipeline Delay non-linearly decreases and the Gate Number linearly increases with the number of stages, respectively. The following Trade-Off function was defined:

$$Trade-Off_k = \frac{1}{(1-P)*NormGates_k + P*NormDelay_k}$$

In which the k parameter is the number of pipeline stages [1, 2 .. 10] and the P parameter [0 .. 1] is the normalized importance given to the speed of the system (1-P is the importance given to the hardware cost). In this design the speed of the system was given a weight of 60 %, and consequently the hardware cost weight was 40 %.

$$Trade-Off_k = \frac{1}{0.4*NormGates_k + 0.6*NormDelay_k}$$

When the Trade-Off expression is plotted for both Standard PFPU and DSP PFPU modes against the Number of Stages (k), *the maximum is achieved for k=7.* Accordingly, this is the number of stages chosen for the PFPU design. From this data it is possible to calculate the maximum speed of the PFPU, that will be the inverse of the time taken by the critical stage plus the time taken by the storing cell. If the 130nm-technology is used (same as Pentium 4), that has a delay time of Td= 1.5ps, the theoretical speed that can be achieved is:

*1/(number of transistors in the critical path * Td) =*

9.009 GHz for the Standard PFPU, and

26.975 GHz for the PFPU in DSP mode.

Dissipated Power

The dissipated power is a very important characteristic of the PFPU, since it can be the discriminating factor when it comes to deciding if the PFPU can be implemented for portable systems. When the number of pipeline stages is increased, more logical gates are inserted in the circuit and more power is dissipated each time this gates commute at the clock frequency.

The analysis of the dissipated power of the Adder, Multiplier and Divider blocks as a function of the pipeline stage number indicates that the power consumption increases linearly with the number of stages but only slightly. On the other hand, if a large sequence of the same type of operation is issued to the PFPU, only one operation block would be really active. Therefore it could be convenient to turn off the idle blocks to save power (interesting feature for portable systems).

The Figure 6(a) shows the PFPU Power Dissipation ratio for single-operation sequence executed in bursts when using the Power-Save control. The Figure 6(b) shows the PFPU Power Dissipation ratio for alternant single-operation sequences executed in bursts when using the Power-Save control in both Standard and DSP PFPU modes. The PFPU has 7 stages.

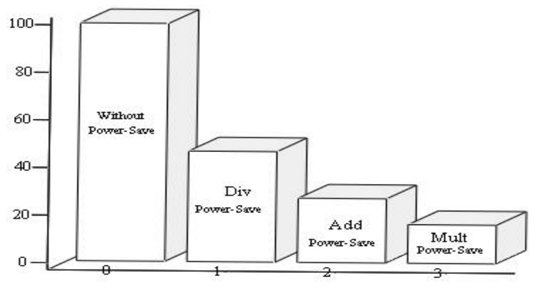


Figure 6(a):
PFPU power dissipation with Power-Save for single operation executed in burst compared to PFPU without Power-Save module.

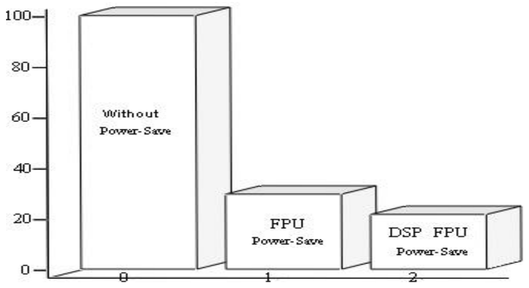


Figure 6(b):
PFPU power dissipation while running burst operations supposing that each operation runs 33 % of the time for the PFPU and 50 % of the time for the DSP mode PFPU

The dissipated power depends mostly on the technology used for the PFPU (parasitic capacity of the MOS transistor), the square of the supply voltage, the operating frequency and the number of transistors that are currently working ([7], [8]). If this design were physically implemented today, the technology to use would be 130nm (same as for Pentium IV and Athlon XP processors), which are the top of the line processors for the PC family. In order to get reliable data for this technology, a simple approximation was chosen, based on the actual Pentium IV datasheet:

Pentium IV: Frequency: 2.5 Ghz; Power: 54.7 W; Supply Voltage: 1.5V; Number of Transistors: 55 million).

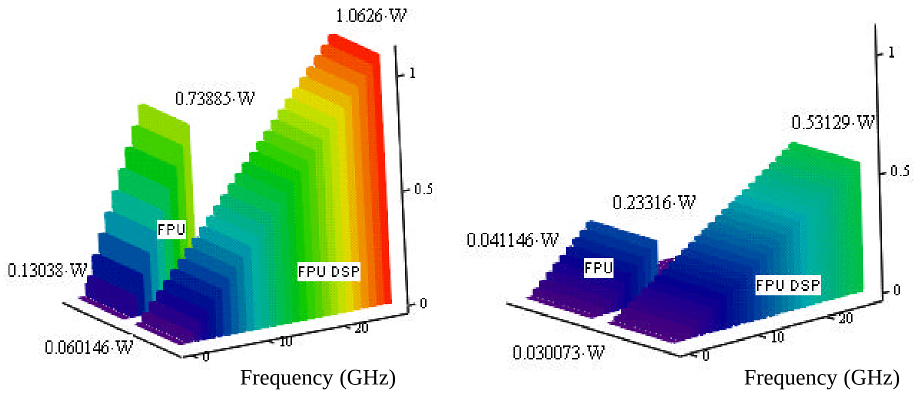
From this data it is possible to calculate a Power Factor (PF) parameter for the Pentium IV processor:

$$PF_{P4} = \frac{Power}{Freq * NrTrans * Vdd^2} = 1.778 \times 10^{-7} \frac{Watt}{GHz * Trans * V^2}$$

This factor will be used to calculate the worst case power dissipation for the PFPU, involving all transistors of the corresponding stage in operation.

PFPU: Frequency= 1.5GHz; Number of Transistors: 130,000

The Power Consumption of the PFPU as a function of frequency is presented in the Figure 7.



Dissipated power for non-burst operation. Power-Save Deactivated.

Dissipated Power for burst operation. Power-Save Activated.

Figure 7: Power Consumption for PFPU Non-Burst and Burst Operations

The Pentium IV 2400-MHz processor has a 2.933 GFLOP floating point operations throughput, dissipating 54.7 W. The PFPU has a 2.5 GFLOP floating point operation throughput at a frequency of 2.5 GHz (due to the pipeline), dissipating 41mW. The ratio between Calculation Power and Dissipation Power in the PFPU is approximately 680 times higher than the one in Pentium IV. Of course, this should be taken as reference only, since it is a comparison between a FPU and a processor.

3. Simulations Performed

While simulating the system, several effects concerning the physical structure of the logical gates were taken into account, including them into OrCAD´s PSPICE parameters in order to guarantee a better approximation to real-world designs. This effects will be described below.

Delay time

It is the time it takes the signal to propagate through the logical gate and present a stable value at its output. The following table shows the delay time of different gates.

Due to a limitation in PSPICE time scale resolution, picoseconds are not correctly managed, hence it was decided to scale up from picoseconds to nanoseconds. This consideration only affects the way the scales are read: nanoseconds has to be interpreted as picoseconds. This consideration should be taken into account to understand the Figure 8, in which different gates were excited at time T0 to evaluate their delay times.

Logical Gate	**Maximum Time [ps]**
AND	4,5
OR	4,5
XOR	4,5
NOR	3
NAND	4,5
INV	1,5
INVTL	6
INVTH	6
BUF	3
BUFTL	6
BUFTH	6

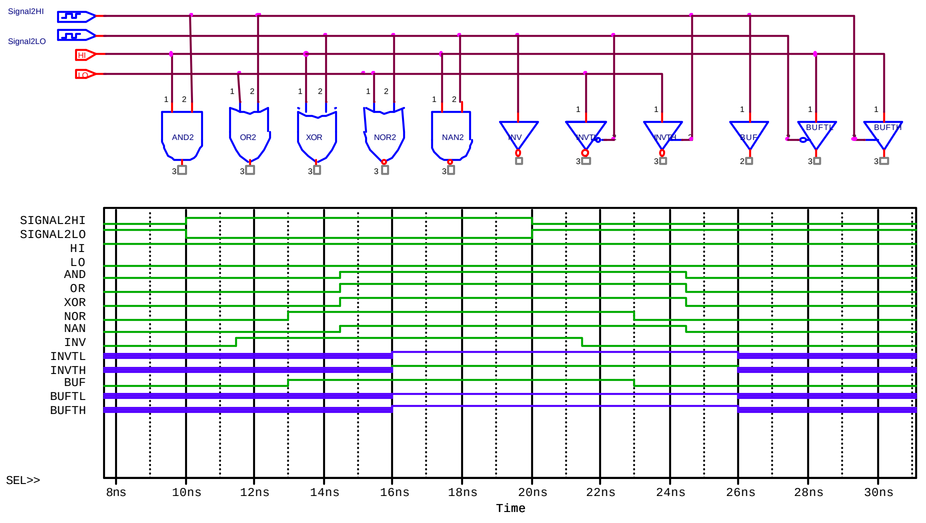


Figure 8: Simulated Gate Delays

Energy

All digital gates requires a minimum quantity of energy applied to their inputs to achieve a stable state at their outputs. Since this energy depends on the voltage and duration of the input signal, and the voltage is fixed by the technology used, the controlled parameter is the time the signal is present at the gate input. If this time is not large enough then the gate will produce no change at its output. As long as the energy at the input increases (the input signal lasts longer) more gates are recruited for operation as their minimum requirements of energy are fulfilled.

The most time-restrictive stage in the PFPU is the divider, which uses 61*Td. This value must be added to the 10*Td, the time taken by the latch to ´remember´ the output data. Using 130-nm technology, the critical time is Tc= (61+10)*1.5ps= 106.5ps. In order to get a more robust system, a longer delay was considered for Tc. The final choice was 120 ps, that produces 8.3 GFLOPS of calculation power.

Input Data

To generate the input data for this simulation, an ANSI C program was designed, to generate floating point random numbers and operations that change every 10 clock cycles. This program output is then fed into another program input. The purpose of this new program is two fold:

- to format the data so that OrCAD can understand it, and
- to execute the operations listed by the first program so the results obtained in the OrCAD simulation can be checked for correctness.

Since the data is given in both decimal and hexadecimal format, this second program also implements a conversion algorithm.

Output Data

Pspice simulation results are stored in a .DAT format, in which all signals are stored. Since this format is coded Pspice interface was used in order to study this signals and export them to an hexadecimal format file. Using this two methods it was possible to verify that the output of the designed hardware was correct compared to the output of the program written in ANSI C.

4. Results

The comparison of a single PFPU against a processor is not fully fair, since the processor has many more instructions which generally lowers its performance. Unfortunately it is hard to find specific information concerning commercial processor's ALUs. Therefore, the following comparisons should be considered as rough approximations of the performances achieved by the PFPU designed. In order to make the comparisons as realistic as possible, the following criteria were considered:

1st Criterium

All comparisons are done following the same path as the processor manufacturer.

E.g. 1: If a processor with an clock frequency "X" doubles it internally, then the comparison is done considering a PFPU running at a clock frequency 2*X.

E.g. 2: If a processor uses superscalar technology that implements 2 pipelines with 2 ALUs inside them, then the comparison is done using 2 parallel PFPUs running at the same clock frequency.

2nd Criterium

Three plots are presented:

a. Processing speed of the commercial processor and processing speed of the PFPU following the manufacturer's criterion

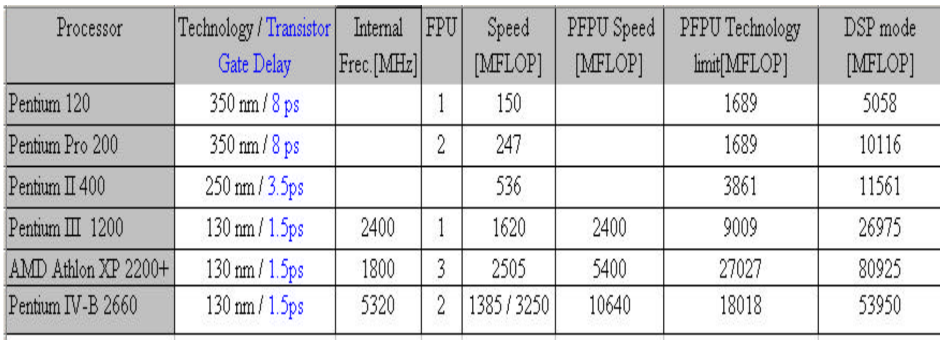
b. Maximum processing speed of the PFPU allowed by the technology used.

c. Maximum processing speed of the DSP mode PFPU allowed by the technology used

3rd Criterium

Burst operation is assumed for speed calculation. This is valid when the circuit has been running for long time (initial and final delays are very small in comparison with the operation time).

The table of the Figure 9 summarizes the comparison between several processors and the PFPU in different scenarios.



Processor	Technology / Transistor Gate Delay	Internal Frec.[MHz]	FPU	Speed [MFLOP]	PFPU Speed [MFLOP]	PFPU Technology limit[MFLOP]	DSP mode [MFLOP]
Pentium 120	350 nm / 8 ps		1	150		1689	5058
Pentium Pro 200	350 nm / 8 ps		2	247		1689	10116
Pentium II 400	250 nm / 3.5ps			536		3861	11561
Pentium III 1200	130 nm / 1.5ps	2400	1	1620	2400	9009	26975
AMD Athlon XP 2200+	130 nm / 1.5ps	1800	3	2505	5400	27027	80925
Pentium IV-B 2660	130 nm / 1.5ps	5320	2	1385 / 3250	10640	18018	53950

Figure 9: Performance comparison between the PFPU and commercial processors

The Figures 10, 11, 12 show the Calculation Power Ratio between the PFPU and commercial processors in three scenarios: all processors running at the same frequency, PFPU running at the maximum speed and the PFPU running in the so-called DSP mode (Divider function removed), respectively. The Calculation Power of commercial processors was measured using *Sisoft Sandra* software ([9]). The Calculation Power of the PFPU was obtained by simulating the design with OrCAD, MatLAB and MathCAD commercial software. In these simulations it was assumed that operands were available 100% of the times (best case) and real world conditions, such as jump prediction, interrupts or the like were not taken into account.

🟦 Commercial Processor	🟧 PFPU at frequency limit
🟨 PFPU at the same frequency	🟥 DSP mode PFPU

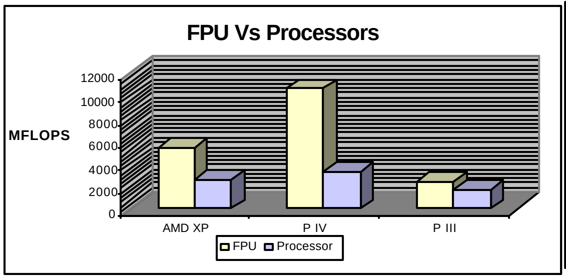


Figure 10:
The Calculation Power Ratio between the PFPU and commercial processors running at the same frequency is:

- Pentium III: **148 %**
- Pentium IV: **227 %**
- AMD XP: **115 %**

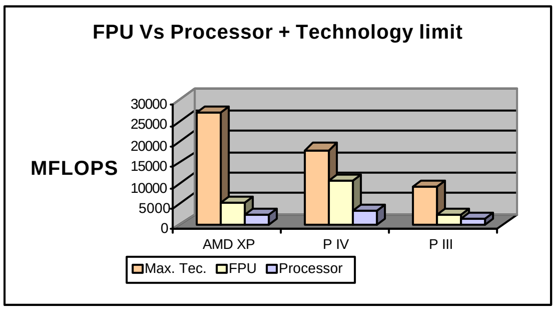


Figure 11:
When the clock frequency is taken to the maximum value allowed by the technology, the Calculation Power Ratios between the PFPU and the commercial processors are:

- Pentium III: **556 %**
- Pentium IV: **554 %**
- AMD XP: **1789 %**

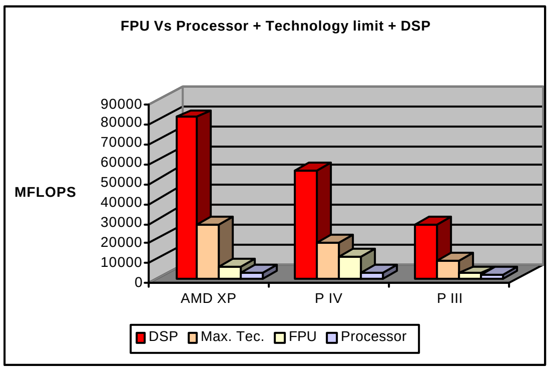


Figure 12
If the Divider function is removed from the circuit (DSP mode PFPU), the new Calculation Power Ratios are:

- Pentium III: **1660 %**
- Pentium IV: **1660 %**
- AMD XP: **3230 %**

5. Conclusions

In this work the design and simulation results for a seven-stage Pipelined Floating Point Unit (PFPU), that complies the IEEE 754 standard for single precision operations, is presented. To reach this objective, basic calculus algorithms were used to solve mathematical operations. Since these algorithms were not absolutely efficient to achieve a high operation throughput, the use of a pipeline was introduced in the architecture to enhance its efficiency. By performing structure comparative analysis the problem was focused to obtain high floating point operation rate (in comparison with top-of-the-line commercial devices) with low hardware cost (evaluated as the number of gates involved). The simulations performed confirmed that the system behaves correctly under all circumstances and that the efficiency obtained was the expected one.

The pipeline enhancement of the proposed PFPU is capable of increasing the operation flow up to 600%. This fact allows a system comprising non-optimized blocks to reach operation processing speeds of at least 50% larger than the top-of-the-line devices such as Pentium IV processors. The PFPU can cover a wide range of applications, from integrated portable systems, for which the Power-Save feature diminishes (48% - 82%) the dissipated power when running operation bursts, to high calculation power applications, where the ratio between Calculation Power and

Energy Consumption is larger than in commercial devices. These figures should not be considered strict but illustrative, because the comparisons were performed between a single PFPU and complete commercial processors, due to the lack of available information regarding the internal modules of the latter.

Since it was proved that the divider block limits the PFPU performance, a special operating mode for the unit was envisaged. This mode does not use that block and is called DSP (Digital Signal Processor) mode, since DSPs are optimized to run operations in the form A*B+C*D. Under this condition, the simulation results indicate that the PFPU Calculation Power (measured in MFLOPS) could reach unmatched values, equivalent to that of eight parallel-working Pentium IV 2600-MHz processors. The main characteristics of the PFPU are listed below. Further tests have to be performed to confirm these preliminary outcomes.

Calculation Power:
up to 8 times (DSP mode) the one obtained from a Pentium IV 2600MHz with the same technology.

Power Consumption:
Standard PFPU: From 41 mW (@ 1,5 GHz) up to 233 mW (@ 9 GHz)
DSP: From 30 mW (@ 1,5 GHz) up to 531 mW (@ 30 GHz)
(Reference: Pentium IV Processor dissipates 54700 mW @ 2,4 GHz)

Power-Save system: From 48% up to 82% energy saving for burst operation.

Pipeline : 600% increase in operation throughput

6. References

[1]: Intel, ***"Pentium architecture FPU"***, **http://www.intel.com/design/intarch/techinfo/Pentium/PDF/fpu.pdf**

[2]: "***IEEE Standard for Binary Floating-Point Arithmetic***", ANSI/IEEE Standard 754, 1985.

[3]: D. Goldberg, "***What Every Computer Scientist Should Know About Floating-Point Arithmetic"***, Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto.

[4]: W. Kahan, "***IEEE Standard 754 for Binary Floating Point Arithmetic, Lecture notes on status of IEEE 754"***, May 31, 1996.

[5]: W. Stalling, "***Organización y Arquitectura de computadoras: Principios y aplicaciones"***, Ed. Megabyte, 1995.

[6]: A. S. Tanenbaun, "***Structured Computer Organization",*** Fourth Edition, Prentice Hall,1999.

[7]: E. Jacobs, "***Using Gate Sizing to Reduce Glitch Power",***

[8]: L. Seed, "***Power Consumption in Circuits"***, University of Sheffield E.E.E. Department Electronic Systems Group, Low Power Circuit Design Course 25 May 2001.

[9]: Sisoft Sandra, "***Benchmarking Software"***, www.sisoftware.demon.com.uk