Computing Polynomials

Alexandre David
B2-206
The Problem

- How to compute efficiently polynomials of the form
  \[ \sum_{i=0}^{n} a_i x^i \]

- Naïve approach: Compute each term and sum-up. If naïve power algorithm is used: \( n(n+1)/2 \) multiplications + \( n \) additions.

- Better?  

Idea

- Rewrite the polynomial as:

\[
\sum_{i=0}^{n} a_i x^i = a_0 + x(a_1 + x(a_2 + \ldots x(a_n)\ldots))
\]

- Compute “a” as:

\[
a = a_n \\
a = a_{n-1} + x*a \\
\ldots \\
a = a_1 + x*a \\
a = a_0 + x*a
\]

*Horner’s rule*
Algorithm

\[ a = a_i[n] \]
\[ \textbf{while} \ (n>0) \ \textbf{do} \]
\[ \quad n = n - 1 \]
\[ \quad a = a_i[n] + x*a \]
\[ \textbf{done} \]
\[ \textbf{return} \ a \]

Notes:
- Efficient if most \( a_i[n] \) are not null.
- Not necessary the best precision.

- Optimized for some DSPs that have a \textit{multiply and accumulate} instruction.