IMPROVED FUNCTION GENERATOR FOR DEVICE CONTROL FOR THE GSI CONTROL SYSTEM

Stefan Rauch, Marcus Zweig, GSI Darmstadt, Germany

Abstract

In the GSI control system [1] a function generator (FG) is used to control equipment with timing functions (ramps). It is situated between the real-time equipment controller (EC) and the actual device control electronics. It provides a 24 bit wide output with an internal accuracy of 32 bits. In ramping mode the FG is configured from the EC with interpolation points. By interpolating the function values the communication on the field bus is minimized. Presently, the interpolation in the FG is linear, which requires only one accumulator of 32 bit width. To better fit the physical functions with less interpolation points we have extended the generator to quadratic interpolation implementing a 2-dimensional arithmetic progression algorithm. This is realized with a datapath of two accumulators. The system should be able to use the complete dynamic range of 2^{15} bits (signed) within one interpolating interval. To meet these requirements the input has to be shifted and the internal accuracy of the datapath has to be 40 bits. Simulations of the datapath have shown that although the accumulators use more resources, the system performance requires only a low cost FPGA like the Altera CycloneII.

WHY A NEW FUNCTION GENERATOR?

In the present system we use a linear interpolation for curve fitting. This naturally leads to an systematic error, depending on the number and distance of the interpolation steps. Every interpolation step needs an parameter, which has to be transferred over the MIL-STD-1553B field bus. Its bandwidth is limited to 1Mbit. Because there are several devices on the field bus, which all requires bandwidth, the basic idea is to minimize the data sent over the bus. Ideally this must not lead to an greater interpolation error. Another idea is, to improve the quality of the fitting. The problem here is, that with the actual design this is contrary to the bandwidth limitation.

REALIZATION

The trivial approach for improving the interpolation is to use more interpolation steps. The problem is, with the MIL bus, every 16 bit telegram needs 20 µs for transmission. With the highest interpolation frequency of 2048 kHz and the shortest distance of 256 steps between interpolation points, the current FG is limited to 6 devices on the bus. Our approach is now, to use a quadratic interpolation instead. So we need fewer interpolation points with the same interpolation error.

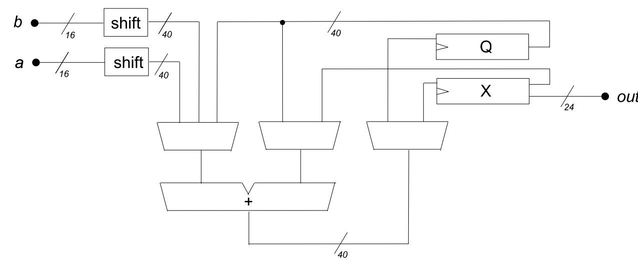

Figure 1: Scheme of the Datapath

Table 1: Values for $a = 2$ and $b = 4$

Q_n	4	8	12	16	24	28	32
x_n	6	16	30	48	70	96	126

INTERPOLATION WITH DISCRETE STEPS

The idea for mapping a quadratic function in discrete steps is to express the function values as values of an arithmetic progression. This can be written with the formula:

$$Q_0 = 0, x_0 = 0$$

$$Q_{n+1} = Q_n + b$$

$$x_{n+1} = x_n + a + Q_n$$

where Q is the quadratic term, a the linear increment, b the quadratic increment and x_0 the start value of x. The relation to a function in normal form can be described as follows:

$$f(N) = \underbrace{\frac{1}{2}b}_{A}\ N^2 + \underbrace{\left(a + \frac{1}{2}b\right)}_{B} N + \underbrace{x_0}_{C}$$

So from the function in normal form ($A * N^2 + B * N + C$) the parameters for the function generator (a and b) can be calculated with $b = 2 * A$ and $a = B - A$ and $x_0 = C$. For using this in a digital datapath, the design needs two accumulators. To calculate the next function value the quadratic term (Q_n) is incremented and then added to the linear term. The complete operation needs three additions an two registers. At the end the function value is in register x_n. For the values $a = 2$ and $b = 4$ you can find an example calculation in Tab. 1 and the relating graph of $f(N)$ in Fig. 2.

CONSTRAINTS

When it comes to build the FG in hardware, there are a few constraints, which influence the numeric behaviour.

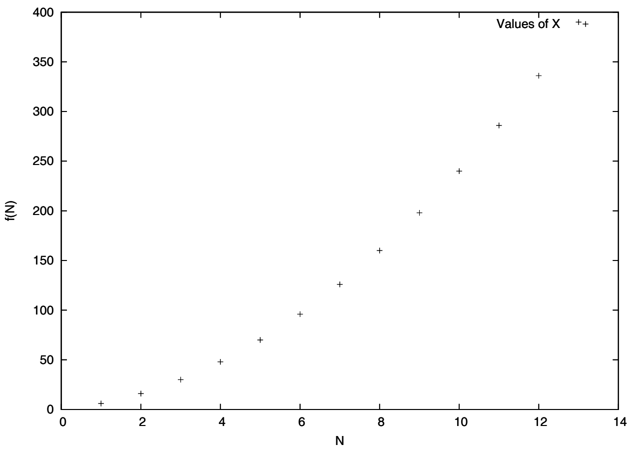

Figure 2: Function values of register x

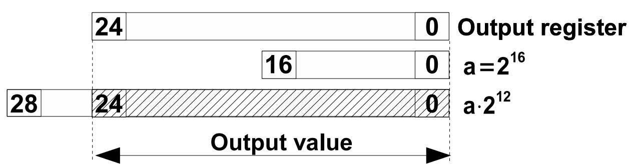

Figure 3: Without scaling of a

The function generator has a 24 bit wide output. The parameters a and b are defined as 16 bit wide, based on the message size of the field bus. We distinguish between the constraints for the linear and the quadratic parameter. With the linear parameter the generator should be able to use the full dynamic range of 2^{24} bit in one interpolation point. With 2^M being the distance between two interpolation points and 2^{16} being the dynamic range of the parameter, the relation can be described as $2^{16} * 2^M \leq 2^{24}$. So the parameter has to be right-shifted $M = 8$ bits. To not decrease the resolution of 16 bit, the internal accumulator has to be bigger than 24 bit (24 + |max shift size|). Fig. 3) shows the arithmetic overflow, which occurs if the parameter a is not scaled.

Quadratic Term

The quadratic term $Q(N) = \frac{1}{2}N(N+1)b$ can be simplified as $\frac{1}{2}N^2b$ for $N \gg 1$. The constraints should be the same as for the linear term, so the full dynamic range could be reached within one interpolation point. Because

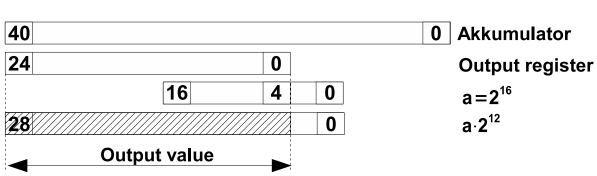

Figure 4: With scaling of a

the quadratic term is more likely to overflow, the derivation of the scaling factor has to consider two cases. In the first case the least significant bit of the 24 bit output should be reached within one interpolation point. For the parameter $b = 1$ and $M = 10$ the scaling factor could be described as:

$$x = \frac{1}{2}N^2 * 1 = \frac{1}{2}2^M * 2^M * 1 = \frac{1}{2}2^{2M} * 1 = 2^{2M-1} * 1$$

$$\Rightarrow M = 10, x = 2^{19}$$

So for 2^M interpolation steps the parameter b has to be shifted by 19 bits.
The second case considers the arithmetic overflow for larger values. For $b = 2^{16}$ the derivation is:

$$x = 2^{2M-1} * 2^{16}$$

$$\Rightarrow M = 10, x = 2^{19+16} = 2^{35}$$

So the parameter b has to be shifted by 11 bits.

SOFTWARE INTERFACE

The FG needs several parameters at the initialisation and prior to every interpolation point. At the start the value x_0 has to be delivered 24 bit wide. There are 8 registers which can be configured at the start with the number of interpolation steps. So 3 bits are needed for every interpolation point, to choose the value M. The parameters a and b are expected 16 bit wide. The amount for shifting parameter a is depending on the value M. Because the scaling of parameter b depends on its size, the shift value has to be delivered too.

CONCLUSION

The quadratic interpolation helps to better fit the given function. Therefore the distance between interpolation points could be greater. The field bus benefits from the lower data traffic too, so more devices can be controlled over the same bus. Another benefit is the decreased load for calculating real time data.
On the other side, the processing of the non real time parameters becomes more complex, because the parameters and scaling values can change with every interpolation point.

REFERENCES

[1] U. Krause, V. RW Schaa, "Re-Engineering of the GSI Control System" (WEAT002), Proceedings of ICALEPS 2001, San Jose.

[2] P. Kainberger, "FG – Gerampte Geräte mit Funktionsgenerator in SIS und ESR", `http://bel.gsi.de/mk/fg/gm_fg.pdf`