THE BRISC: A DIGITAL SIGNAL PROCESSOR DEDICATED TO THE COMMAND OF AC ELECTROMECHANICAL ACTUATORS

ERIC GILSON dr ir
SABCA (Société Anonyme Belge de Constructions Aéronautiques)
1470 Chaussée de Haecht
B-1130 Brussels
BELGIUM

The BRISC (for Biprocessor RISC) is an original integrated circuit that represents a new approach for actuator control: the BRISC is a DSP (Digital Signal Processor) that has been designed aiming specifically at the local command of AC motors and EMAs (ElectroMechanical Actuators). DSP is attractive for its flexibility in control algorithms and its ability to perform intelligent tasks, like self-commissioning and monitoring. The BRISC, being specific, is both low cost (it fits a 64 pins cerquad) and very efficient (it is a vectorial number cruncher with an integrated PWM -Pulse Width Modulator-). Moreover, it can be processed according to MIL-STD-883C (group B) and CECC 90.x00 (up to CCQ Y, class B). The paper presents the BRISC and a case study of the command of an EMA for thrust vector control. The EMA is made of a BLCD (Brusless DC Motor), with its resolver, a ball screw and a lvdt. In this application, the BRISC handles vectorial AC motor control, position control and digital signal conditioning of the two position sensors.

Introduction

Why DSP for motor control?

- *for a high quality motor drive*

Space and airborne systems require the best motor current control. The best means :
- minimum motor losses
- maximum performance (speed, torque)
- limited EMIs
- smooth operation

With DSPs, the control can be fully adapted to any motor because it both makes possible the implementation of sophisticated control schemes and does not require a completely new hardware for each motor to be controlled. And one knows how motors can be different: just consider the back EMF profile or the cogging torque. Besides, the development of an optimum command is considerably eased by the flexibility of software driven systems.

- *To put intelligence in the actuators*

If a DSP controls an actuator, it is best suited to perform monitoring of the actuator because it is intelligent enough to run an actuator model and because it naturally knows all the internal variables of the system -like the motor currents and voltages-. Those are of interest because they permit the fastest detection of an actuator malfunction. To illustrate this, let us consider a flight control actuator. Usually, the monitoring is performed by the flight control computers (FCC). Due to the FCC computational load and to the bus work load, fault detection can only be performed on a "slow" variable: the actuator position which is compared with the order, with a time delay and a threshold. This means a slow detection that leads to a surface hardover when switching from the nominal actuator to the redounded one. The job can better be done by the DSP that controls the actuator. That DSP can perform fast detection and send just a digest status to the flight control computer.

- *To take advantage of the noise immunity of the digital systems*

- *And probably much more to be discovered...*

Why the BRISC ?

SABCA is concerned in the development and fabrication of military and civil, ground, airborne and space electronics, devices, sub-assemblies and actuators. As the work on EMAs and EHAs (ElectroHydrostatic Actuator) was progressing, the actuator service realized that a specific DSP for their command could bring an important asset to the smart actuator concept. That DSP is the BRISC, for Biprocessor RISC. Its dedicated nature brings benefits over other processors :

- *The BRISC is built above PWM*

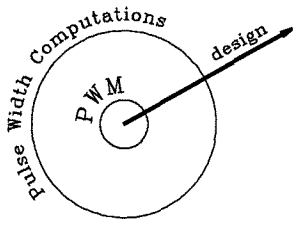

Figure 1 history of the BRISC

The PWM is fundamental in motor control and should not be regarded has a peripheral function, because it is the ultimate link in the control chain. The BRISC design is based on a previous ASIC that performed only PWM[1]. In the BRISC, the PWM timers are accessed as general internal registers and the state of the main timer is reachable like any flag of the arithmetic units. As a result, the generated PWM waves have no distortion due to uncontrolled timings.

Besides, all the computations and data acquisitions can be synchronized on the PWM carrier which is very valuable for:

- general noise immunity during analog data sampling.
- resolver and LVDT conditioning, because, as the BRISC can easily generate the sensor excitation through one of its PWM channels, it can use the PWM carrier as a reference to select the right sampling moment within the electrical period of the sensor.

- *The BRISC features a specific proprietary DSP core*

The specific DSP core has two benefits:

- **a very high processing power** that proved to enable the full software implementation of most of the functions required in modern AC servo control: an application is reported hereunder (a linear EMA control) that features 26KHz motor PWM, 6.5Khz current sampling, resolver *and* LVDT excitation and conditioning. The processing power comes from :
  - a dual datapath to exploit the intrinsic parallelism of the applications (the experienced parallelism is in the area of 40%),
  - an important register structure (the experienced efficiency is two data access in registers for one in memory)
  - a light weight floating point , 16+8 bits, and a simplified RISC control that allow short clock cycles.

- **a low cost** : only what is known to be necessary is implemented. As a result, the BRISC, despite its power, is a small chip (65 mm^2 in a CMOS 1 micron technology) with a low pin count (it fits a cerquad or PLCC 64 carrier).

The main paths from application considerations to the BRISC architecture are given on figure 2.

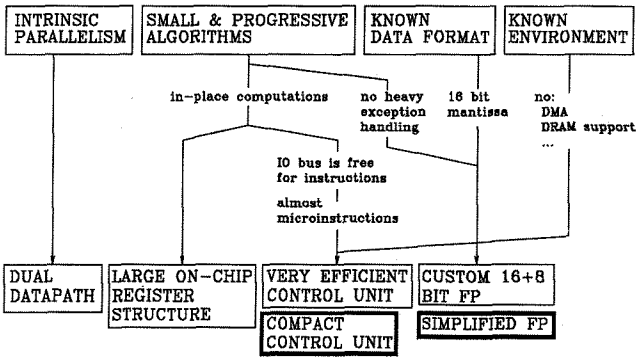

Figure 2 paths to the BRISC architecture

The most obvious characteristic of the BRISC is its dual datapath that allows it to perform vectorial computations. The choice of a vectorial datapath originates in the vectorial control of AC machine: the main variable in AC machines is the rotating magnetic field. The control technique described hereunder in the case study exemplifies the vectorial nature of the system.

One important key to efficiency and compactness is the control unit. The BRISC control unit is extensively described in [2]. Due to the fact that the applications are known to be quite small and that the code is mostly streamlined (most of the time, the processor travels one or more endless loops, with a fixed time scheduling), it could be assumed that most of the data can be located in the internal register structure. Hence, the I/O being almost dedicated to instruction fetch, the BRISC can live with a heavy instruction transfer rate. The BRISC executes a code that is very close to a normal machine microcode: all the instructions are single cycle, except for the floating point instructions. The control unit is both simple (low area) and fast: no logic is required for the synchronization of the stages of the instruction pipe-line (fetch-decode-execute). Besides, an instruction set of such a low level let a maximal freedom to the programmer for code optimization: any cycle of pipe-line latency can potentially be used for a useful instruction. The instruction set is also modular. For instance, the floating point resources can be used independently from the floating point instructions, allowing fixed point arithmetics on the same functional units.

The floating point feature of the BRISC is also of importance. We did not conform to a 32 bit standard which would have been too expensive for such a dual datapath machine (there are two floating point units working in parallel). Instead, a reduced format has been selected: 16 bit mantissa and 8 bit exponents. The 16 bit mantissa are sufficient for the considered applications and are

intrinsically faster than 24 bit mantissa. A figure of merit of the floating point units is the peak MFLOPS figure: it reaches 33 MFLOPS with the current version of the chip (1 micron, 60ns cycle).

Description of a practical implementation

The system

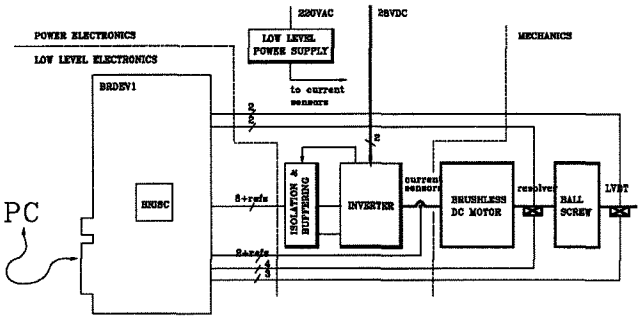

Figure 3 : System description

BRDEV1 (BRisc DEVelopment 1) is an evaluation and development board for the BRISC for EMA and possibly EHA control. The card fits the ISA bus (16 bit) of a PC or compatible machine. In the presented application, only the power related devices are outside BRDEV1, in this case, the conditioned power supplies, the voltage inverter and the current measurement devices (zero flux transformers).

The commanded actuator is an EMA made of a brushless DC motor with its resolver that drives a ballscrew. An LVDT gives the linear position of the actuator.

The motor is 3-phase brushless DC, with a 16 pole rare-earth permanent magnet rotor. It is rated at 20 A_{rms} max. under 28 volts. Its max. speed is 550 rpm. The motor drives directly a screw ball. The inverter is made of discrete MOS transistors. The gates are commanded through floating supplies and opto-isolators.

The hardware: BRDEV1

The resolver and LVDT do not require specific converters to digital: the raw signals are converted by a general purpose multiplexed ADC (Analog to Digital Converter) system that is used for resolver, LVDT, currents and also orders that come in this case from an analog signal generator.

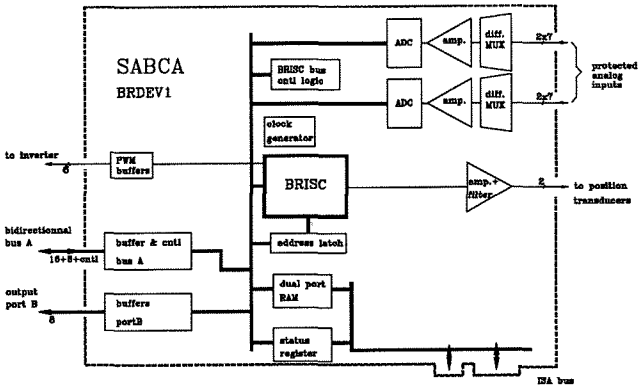

Figure 4 : BRDEV1

The hardware support implemented on BRDEV1 is the following:

- *Direct BRISC support*

This includes :
- Program and data RAM, implemented with dual port SRAM (32bitx16K module). The second port of the DP RAM is dedicated to the ISA bus (the PC down loads code and fetches BRISC reports -a trigger is implemented-).
- Address latching and data bus buffering for the peripherals.
- Peripheral address decoding
- Some glue logic.

- *Signal preconditioning and multiplexed ADC system*

The ADC system feeds the BRISC with sampled data from the sensors. The analog inputs are voltages in the range -5v to +5v. If required, signal pre-conditioning (like pull-down resistors and low-pass filters) can be implemented with piggybacks cards.

The inputs are differential. Two channels can be sampled at a time, like :
- two currents (i.e. two LEM outputs)
- two resolver secondary voltages
- two LVDT secondary voltages.

The ADCs are sampling, 12bit, 100Ksps. BRISC selects the inputs and initiates the conversions. The ADC system is operated in pipe-line (select channel and start conversion, fetch result).

- *Buffer for resolver and LVDT*

The excitation signals comes from the BRISC as PWM waves. They need external filtering and buffering. The two filters are low-pass, fourth order with digitally programmable cut-off frequencies. BRISC selects those

frequencies. Buffering on the card is sufficient to directly drive common synchro, resolvers, LVDT and RVDT.

- *Computer bus interface*

The interface is 16 bit ISA. BRISC is put in reset by the PC on program loading through a dedicated status register on the card. To ensure the integrity of the shared data, the nominal mechanism is semaphore control which is supported by the DP RAM. Mutual interrupts are also possible.

- *Parallel ports (implemented but not used in this application)*

The parallel ports are:
- PORTA : a 16 bit I/O bus (reduced version of an ISA bus)
- PORTB : an 8 bit output port.

PORTB 's primary function is to send logic orders (enables, turn-on, configure) to the external devices (inverter, power supplies).

PORTA can be used either as simple input port or as a bus on which intelligent peripherals dialogue with the BRISC (additional sensor conditioning cards, redounded control system,...).

The software support

The programming language used for developing the presented application is the assembly language. The size of the application make realistic such a programming. The assembler supports structured programming through recursive include files and recursive macros, with parameter passing.

A development environment has been built above the Brief editor. It is menu-based. One interesting feature is the "scope" utility that permits, with the proper code running on the BRISC, to fetch internal BRISC data, to trig on one channel and to display the timing diagram. The test results presented below are produced by that utility.

It is planned to develop a graphical development environment, oriented towards motor, EMA and EHA control.

The application software structure

The fundamental function fulfilled by the BRISC in this application is described in the following figure :

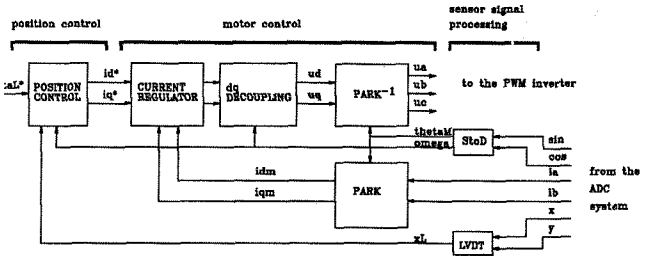

Figure 5 : fundamental function

That type of vectorial motor control can be found in [4]. The used code organization is very simple and illustrates the underlying idea behind the BRISC design.

The control functions are split into elementary functions. Each of the two floating datapath of the BRISC features 8 banks of 8 overlapping register each. The idea is to execute one elementary function in one register bank. The experience shows that one register bank is what is generally required to efficiently implement an elementary function as current control or resolver to digital. Of course, access to memory is always possible. The bank allocation is also made considering that functions that are to be executed sequentially should be in successive banks in order to pass arguments through the shared register whenever possible.

The register usage in the present application is shown in the next figure. Only seven out of eight banks are used: one bank is left free for imagination (model based monitoring?). The bank allocation is identical in the two floating units.

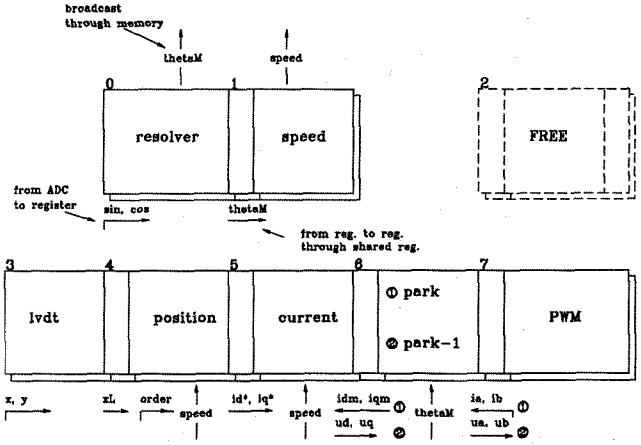

Figure 6 : register usage

The elementary functions as shown in figure 6 match the global function in figure 5 according to:

	becomes
POSITION CONTROL	position
CURRENT REGULATOR & dq DECOUPLING	current
PARK & PARK-1	park/park-1 (in the same bank)
StoD (Synchro to Digital)	resolver & speed (two banks)
LVDT	lvdt
PWM (not shown)	PWM

The global timing is defined. The following timing issues are considered:

- Every action is synchronized with PWM so that each function starts with the beginning of a PWM carrier half-period: the main timer gives one carry-out at

  $$fmain = 2*fc,$$

  with fc the PWM carrier frequency. fmain is the first reference frequency of the system.

- We have chosen to perform resolver and LVDT conditioning in software. This includes PWM generation for sensor excitation. As a result, the PWM carrier frequency is a multiple of the sensor frequency. This frequency, fr, is typically such that

  $$fc/fr = 6..8..$$

  fr is the second reference frequency of the system. Besides, the sensor conversion algorithm require the sampling of the secondary voltages at given instants in the excitation period. Consequently, the resolver and LVDT functions (which include the secondary voltage sampling) have to start on given fmain periods.

- The third reference frequency of the system is the current sampling frequency, fs. fs is usually chosen one order of magnitude beyond the motor's electrical bandwidth. Although there is no direct link between fc and fs, a ratio

  $$fc/fs = ..4..8..$$

  seems practical.

The integration of those three constraints is illustrated on the next figure. The analog input sampling take place at the very beginning of LVDT, park, position and resolver.

We have chosen here,

$$fc \cong 26KHz,\ fc/fr = 8,\ fc/fs = 4$$

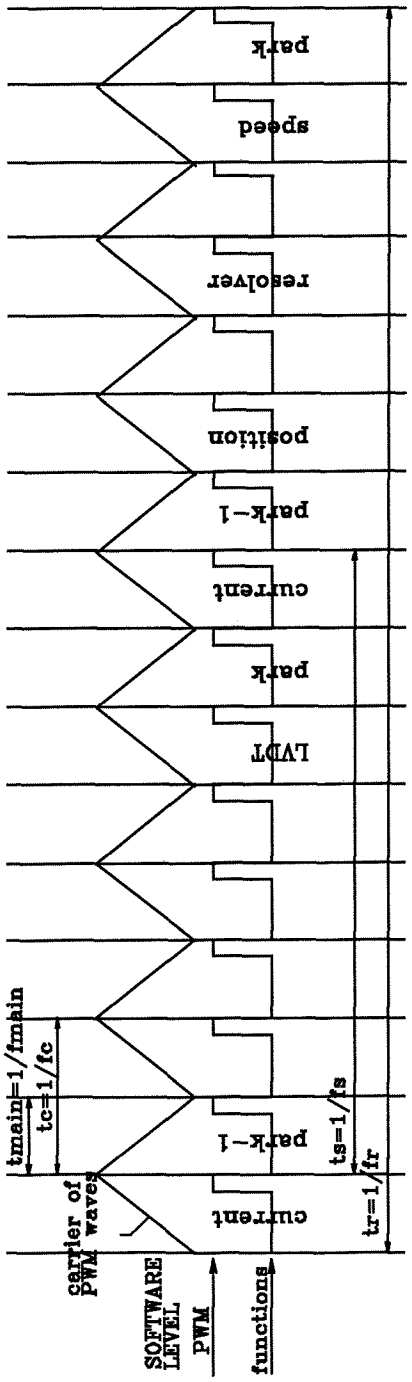

Figure 7 : application timing

There is a jump to the PWM routine at the end of each PWM carrier half period in order to update the content of the output decounters for the next PWM commutation. BRISC then waits for the next main timer period before it starts the next function. For the chosen fc, the execution time of the various routines is generally less than tmain except for resolver that requires two periods. We can see that there is still lot of time available to make use of the free bank registers.

Test results

At this time, the full trial results are not yet available. Nevertheless, sone interesting measurement are shown. In figure 8 : the actuator linear position and the motor angular position during a linear position step . Note that those signal come from the BRISC and not from an external acquisition system. The data are put in the DP RAM by the BRISC, whilst it is controlling the system. Acquisition starts with a trigger on channel one (here the rotor position). The triggering level is defined on-line through the PC interface. The BRISC routine that permits this does not require usage of a register bank. Figure 9 shows the electrical angular position (in radians, between 0 and 16.PI for this 16 pole machine) and the phase one voltage order versus time. The speed is reversed at about 350ms.

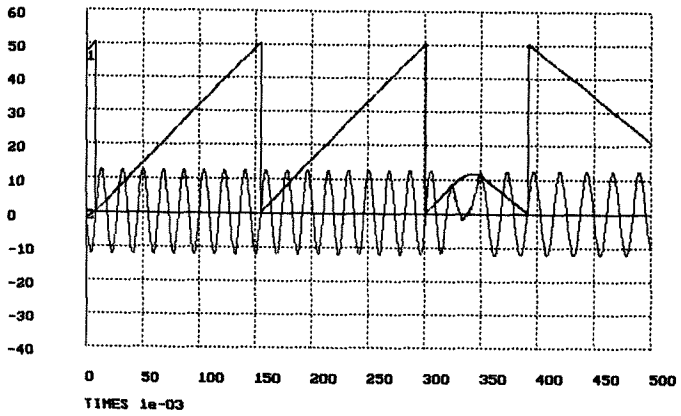

Figure 8 : 1 - Electrical angular position [rad]
2 - Voltage order for phase 1 [v]

Conclusion

An application specific DSP for the command of AC motors has been presented. The laboratory tests performed on a linear EMA are promising. They show that the BRISC is powerful enough to perform a fully digital, software based actuator control: motor control (current loop), actuator position control and resolver (for position and speed) and lvdt conditioning. Moreover, beyond the quality of the drive that has been obtained, it made clearly appear the great flexibility during the development of the actuator control strategies that is enabled by such software based systems. The work is now continuing towards the

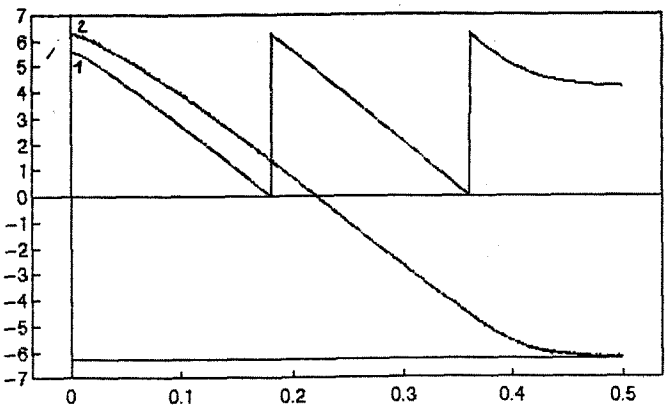

Figure 9 : 1 - Rotor angular position [rad]
2 - Linear position [mm]

exploitation of more of the BRISC resources. The first direction is the control motors that are harder to drive, they include motors featuring an important cogging torque or higher voltage ones. The second direction is the simultaneous monitoring of the actuator by the BRISC, which means running a good motor model in real time.

References

(1) Alain Dumont, Chandra Shekhar, Eric Gilson, Jean-Louis Daout, Jean Remy, KVSH Rao, Paul Jespers, S Srivastava, A PWM processor for three phase solid state inverter control, IMACS proceedings, July 18-22 1988, Paris, France, Vol. 3, pp 151-153.

(2) A. Dumont, E. Gilson, Ch. Trullemans, BRISC: a RISC processor architecture dedicated to power applications,Euromicro '88 proceedings, Aug. 29- Sept. 1, Zurich, Switzerland, pp589-596.

(3) Eric Gilson, Paul G. A. Jespers, The BRISC : a high speed integrated processor dedicated to power control, IECON '90 proceedings, Nov. 27-30, Pacific Grove, CA, USA.

(4) H. Buyse, F. Labrique, B. Robyns, P. Sente, Digital field oriented control of a PM synchronous actuator using a simplified strategy for controlling the Park components of the stator current, IMACS TC1'90 proceedings, Sept.19-21, Nancy, France, pp37-41.