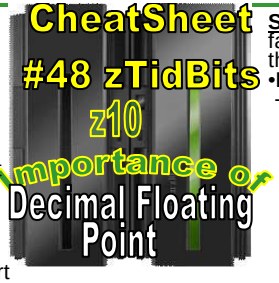


•Did you know that most processors today support fixed-point integers and binary floating point (BFP) in hardware but do not support *decimal arithmetic* natively.
•Decimal arithmetic is very important for "banking" and "commercial financial" applications.
- Even though transistor sizes are shrinking at an exponential rate, today's computers do not support a native decimal fixed-point or floating-point data type, with the notable exceptions of simple pocket calculators and **mainframes**!

•The z10 processor is the ***first mainframe*** with hardware support for the DFP format.
- It joins the IBM POWER6 processor-based System p 570 server as the only hardware support available for this format.
-Commercial applications require a decimal format and will greatly benefit from this new format.
- The direct hardware support for DFP arithmetic is also backed by extensive support in software for these data types. Operating System's, debuggers, programming languages, middleware, and application hosting environments have been enhanced as required to support this new data type.
- This provides our customers with complete stacks to develop and deploy applications that can take advantage of the DFP facilities introduced with the z9 Enterprise Class and Business Class processors and improved in the z10 processor.

•The first hardware implementation of the IEEE 754-2008 floating-point standard using a **DFU** (Decimal Floating-point Unit) was on the POWER6 processor.
NOTE: The z10 DFU is based on this design, and both were implemented by essentially the same team. The POWER6 processor design was initiated first and was generally available in 2007 on the System p 570 server. At first, there was a common design point for the two processors, but it later diverged to meet the unique requirements of the z10 processor.

•The z10 DFU supports DFP operations in a manner similar to the POWER6 processor.
- z10 also supports the traditional fixed-point decimal operations that have been part of the architecture of IBM mainframes for more than 40 years.
- The z10 DFU uses the main dataflow of the POWER6 processor with additional interfaces to the fixed-point unit (FXU) and data cache as well as a completely new set of controls to support the additional instructions in the IBM z/Architecture platform (see POP publication).

NOTE: Both the z10 DFU and the POWER6 processor DFU have 54 DFP instructions, but the z10 DFU has an additional 13 decimal fixed-point instructions and four hardware-assist instructions.

•For fixed-point decimal operations, both source operands and the target are in memory.
☞Most processors today do not support memory-to-memory arithmetic operations because of such difficulties as variable length handling and fetching of memory operands. However, System z processors (CISC) are ***optimal*** for memory accesses because the execution pipeline for one instruction includes both a memory access and an execution stage, whereas POWER*n* (RISC) computers require multiple instructions to accomplish the same task.

SYSTEM z Exploiting DFP

•Software support for the DFP feature was introduced in the System z9 platform with IBM z/OS v1.6 at the time of the initial availability of the z9 feature.
NOTE: The initial support that was provided was limited to the z/OS, z/VM, and IBM High Level Assembler (HLASM). At this time, work to support the z9 facilities has also begun for Linux on System z and the GCC (GNU compiler collection) tool chain.

• **z/OS** supports for DFP was built to utilize the infrastructure provided for IEEE 754-1985 binary floating-point (BFP) standard. This includes detecting the attempted use of **hardware DFP** (**HDFP**) facilities or the additional FPRs[1], activating the hardware features, providing accessible indications of this use, restoring the application state to reexecute the offending instruction, and initiating the saving and restoring of the extended state required by applications that use DFP.

• **z/VM** was updated so that guest OSs could make use of the support provided in the System z9 platform at the initial availability of the hardware, although z/VM itself does not make use of DFP.

•**The Linux on System z** kernel takes a different approach to support. An administrative action to update a file, `/proc/cpuinfo`, is done to indicate a DFP-capable central processing unit, and the feature is always on.

•**Commercial application software** objective for DFP is that it must be possible to support the inclusion and use of HDFP in commercial applications. This led IBM to provide support for DFP data and the hardware DFP instructions in the system components that support the development, deployment, and execution of these applications. Providing appropriate support in the OSs and in assemblers is abase for this objective, but support in other components is required as well.

1. FPR- Floating Point Registers

Software support in z/OS:

Languages and debuggers - The most important programming languages used to write commercial applications that are hosted in z/OS include COBOL, PL/I, C/C++, Java, and HLASM.

☞ **R**educed **I**nstruction **S**et **C**omputing **- RISC** processors only use simple instructions that can be executed within one clock cycle. The fewer, simpler and faster instructions would be better, than the large, complex and slower CISC instructions. However, more instructions are needed to accomplish a task.
Complex **I**nstruction **S**et **C**omputing- **CISC** architecture is to complete a task in as few lines of assembly code as possible. This is achieved by building processor hardware that is capable of understanding and executing a series of operation (ie. Macro). The philosophy behind CISC is that hardware is always faster than software**.**

Software support in z/OS (*continued*) - Support for DFP data backed by hardware facilities is included in z/OS compilers for all of the languages previously mentioned with the exception of COBOL.

•**Datastores** can be divided into those that have schema and those that do not.
- Schema is metadata maintained by the datastore that describes the content of the data. Datastores **without** schema can be used by applications to store and retrieve arbitrary data without enhancement to the datastore itself, assuming appropriate data definitions are available in the programming languages used to implement the applications. This includes floating-point data, which includes DFP data. File systems and datasets supported by IBM access methods can be readily used to store and retrieve DFP data without further enhancement. The same observation applies to the IMS database and the VSAM database supported by CICS. Datastores *with* schema require specific enhancement to the datastore to add support for new data types. This is the case with DB2 for z/OS, a critical component in commercial applications for the System z platform. Such support has been added to DB2 and the SQL (Structured Query Language) used to store and retrieve data.

•**Application-hosting environments-** There are a number of application-hosting environments in the z/OS operating system. Environments provided by IBM can support applications that utilize DFP data and computation. A partial list includes Batch, Time Sharing Option (TSO), UNIX System Services (USS), IMS, CICS, and the Websphere z/OS portfolio.

•**Middleware** DFP data is added as a fully supported DB2 data type, DECFLOAT, which is the built-in data type for z/OS v1.9. DFP data types are defined in SQL for DB2. DB2 supports the 8- and 16-byte data formats, but not the 4-byte format. The Query Management Facility (QMF) products for TSO/CICS and for WebSphere also support the DECFLOAT data type.

•**Language runtime support** IBM Language Environment for z/OS supports the C Runtime Library. A set of intrinsic functions that perform computations using the HDFP instructions hasbeen provided, beginning in z/OS v1.9. The remainder of these functions will be delivered in the future. They will be used and invoked by code generated by the compilers for all of the programming languages except Java. Language Environment also supports the extended state saved and restored by the OS and, for problem determination purposes, provides it to application-hosting environments or directly to problem determination tools. In addition, Language Environment initializes the contents of the FPC (floating-point control) word at the initialization of application programs.

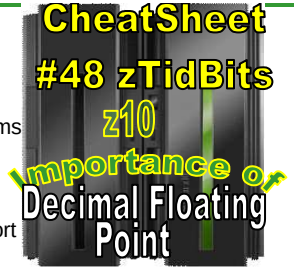
•**Programming languages** The z/OS **C/C++** compilers allow the user to generate C/C++ applications that perform appropriate computations on DFP data using the HDFP instructions included with the z10 processor. The support is delivered with the C/C++ compilers that are part of z/OS v1.9. **Java** supports decimal data through the BigDecimal class library. On a z10 processor, the Java `Just-in-Time` (`JIT`) compiler associated with the Java 6 JVM generates HDFP instructions to implement this class library for existing and new programs. The performance of the parts of an application using the BigDecimal class library running on a z10 platform *will improve* over the same application hosted on an earlier System z machine more than the basic difference between the instruction execution speed of the machines as a result of the use of HDFP instructions. The magnitudes of those improvements are unpredictable and related to the intensity of use of BigDecimal data. The Enterprise **PL/1** compiler has added support for HDFP in release 3.7, which was first shipped in early October 2007. The PL/1 language has had a DECIMAL FLOAT data type since its creation. A new suboption of the FLOAT option is DFPjNODFP. If it is set to DFP, the compiler treats DECIMAL FLOAT data such that it is handled internally using the HDFP data formats and is processed using the new instructions in the hardware. To create programs that take advantage of the new instructions, data formats, and capabilities, existing programs will have to be recompiled using the new suboption and an ARCH level specification of 7. Some new built-in functions have been added and others have been updated. An accommodation is made for some library functions for which an HDFP version has not yet been provided.

•**Debuggers** support for HDFP involves being able to display DFP data stored in main storage. The dbx is the standard UNIX debugger that is shipped with z/OS. It is used to debug programs written in C/C++ or HLASM that are hosted in the UNIX System Services environment within z/OS. Its HDFP support was provided in z/OS release 9. Debug Tool is an optional IBM program product made available as a standalone product or as an option with some z/OS compiler program products. It is the debugger used to debug applications compiled by Enterprise COBOL and Enterprise PL/1. It can be used to debug HLASM and XL C or Cþþ programs. It debugs programs that are compiled with these languages and hosted in many of the z/OS application-hosting environments. This includes batch, TSO, IMS, CICS, and USS (z/OS UNIX).

•**Batch applications** written in HLASM, XL C/C++, Enterprise PL/1, or Java can be written to use DFP data and the HDFP instructions. These programs can use the support provided in DB2 version 9 to store in, retrieve from, and query DFP data in DB2

continued

Continued from previous page.



Software support in z/OS (*continued*) –

- **TSO and USS** applications written in HLASM, XL C/C++, Enterprise PL/1, and Java can be written to use DFP data and the HDFP instructions. These programs can use the support provided in DB2 version 9 to store in, retrieve from, and query DFP data in DB2.
- **IMS** applications written in HLASM, XL C/C++, Enterprise PL/1, and Java can be written to use DFP data and the HDFP instructions. These programs can use support provided in DB2 version 9 to store in, retrieve from, and query DFP data in DB2. They can natively store and retrieve data from IMS databases.
- **CICS** applications written in HLASM, XL C/C++, Enterprise PL/1, and Java can be written to use DFP data and the HDFP instructions if certain compiler options are used.
- **For C, C++, and PL/1 programs**, the compiler suboption AFP(VOLATILE) must be used.
- **DB2**, in addition to its role as a datastore, DB2 hosts applications known as ***stored procedures***. Stored procedures may be written in HLASM, XL C/C++, Enterprise PL/1, and Java to use DFP data. HLASM applications must be Language Environment-enabled to run as stored procedures. Stored procedures hosted in DB2 version 9 can store in, retrieve from, and query DFP data in the relational database. Other datastores without schema can also be used by the stored procedures to store and retrieve DFP data.
- **WebSphere Application Server** applications are Java programs. To the degree that they support decimal data, they do it through the BigDecimal class library. This class library supports decimal data using the HDFP instructions provided in z10 processors when compiled by the Java 6 JIT compiler and the appropriate processor is detected.
  NOTE: WebSphere application programs can use the support provided in DB2 V9 to store in, retrieve from, and query DFP data in DB2 through the use of Sun JDBC (Java DataBase Connectivity) support.
- **GCC stack for Linux on System z** IBM implements support in the Linux kernel and the GCC stack but does not distribute Linux for System z. Distribution is handled by external companies, specifically Novell and Red Hat. The support for the GCC stack described below becomes available when the distributions become available.
  - **Binutils** - The new instructions in Linux were added to binutils 2.18 for support of the assembler, gas, and objdump. This support is expected to become available in Novell SUSE** Linux Enterprise 11 (SLES 11) and Red Hat** Enterprise Linux 6 (RHEL 6) sometime in 2009.
  - **GCC** - GCC added support for the new data types,_Decimal32, _Decimal64, and Decimal128 in a release that was not externalized. The DFP instructions in the z9 processor and the z10 HDFP instructions are included in the GCC 4.3 compiler, which will be shipped in SLES 11and RHEL 6, but are available for download. The options required to use the instructions are -march¼z9-ec and -mhard-dfp.
  - **Glibc** - Glibc implements and contains the intrinsic library functions using HDFP.
  - **GNU Debugger (GDB)** Support for DFP in C has been provided in GDB release 6.8 made available for download in March 2008. GDB 6.8 will be included in SLES 11 and RHEL 6 Linux distributions.

Acknowledgement of this work is to the DFU logic team consisting of Adam Collura, Steve Carlough, Wen Li, and Mark Erle. Also acknowledge the software support leaders Peter Relson, Levon Stepanian, Ian McIntosh,Peter Elderon, John Ehrman, Steve Ball, Jim Shearer, Steve Walkowiak, Francisco Anaya, Chris Crone, EvaHu, Ron Smith, and Michel Hack.