Higher data throughput for DSP analog-to-digital converters

By Heinz-Peter Beckemeyer

Field Application & System Engineer, Data Converters

The processing power of today's digital signal processors (DSPs) goes into the range of several MIPS (million instructions per second). However, these high data rates can be achieved only if algorithms are executed from the internal memory. The maximum communication speed with external peripherals, such as analog-to-digital or digital-to-analog converters, is highly limited due to external bandwidth.

At conversion rates above 1 MSPS (million samples per second), it is frequently difficult to collect the data from the analog-to-digital converter (ADC) in time to process it further.

Data converters with such a high conversion rate most likely have a parallel interface to the parallel data bus of a processor, since the speed on serial interfaces is further limited. The reading of data from such an ADC by a processor normally takes place by means of an interrupt service routine (ISR). The maximum speed, however, is limited, as every processor has a latency time (the time required for first possible access within the ISR). The latency time can add up to some hundred nanoseconds depending on the processor type and its frequency. The resulting speed is thereby limited to 1 to 2 MHz. From this arises the problem of capturing and processing data that is controlled via an interrupt from a fast ADC.

This article suggests a solution that increases the data throughput of an ADC into a DSP by a factor of 5 to 10. This solution is already integrated in one of the latest ADCs, THS1206, developed by Texas Instruments.

Conversion and transfer of data

Fast analog-to-digital (flash/pipeline) converters typically provide a digital conversion value with every clock cycle of the sampling clock.

It frequently becomes a challenge for the DSP to collect the data of these converters and process it at this higher speed. For example, a data converter with a sample rate of 6 MSPS brings a new conversion value to the digital output every 167 ns.

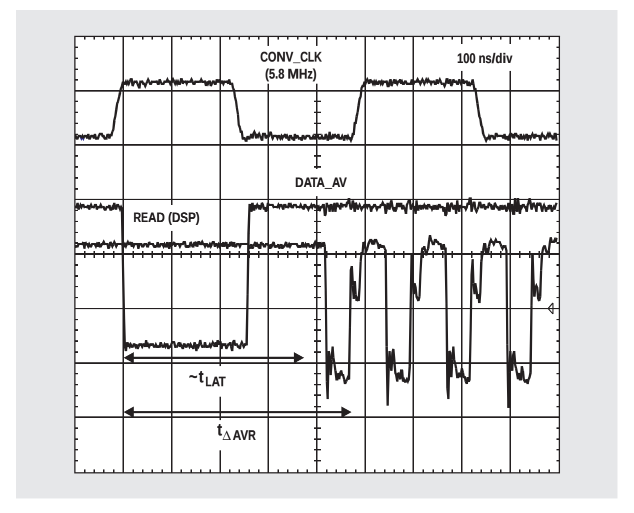
The maximum data throughput is a limitation because of the latency time already mentioned. The latency time of the TMS320C542 DSP with a frequency of 40 MHz is around 400 ns. This latency time is lost every time the DSP jumps into the ISR to read a conversion value.

Higher data throughput by using a FIFO

A higher data throughput can be achieved when the DSP can read a block of conversion values instead of only one value with every jump into the ISR. In this case the DSP loses the latency time only once per block to be read.

Figure 1 shows an example where the DSP always reads 8 conversion values in a block within an ISR, therefore losing the latency time only once per ISR.

Figure 1. Reading a block of data



The latency time can be seen in Figure 1. It is the time from the falling edge of the signal DATA_AV (data available) to the first reading of the DSP.

The reading of the data in a block can be achieved by using a FIFO architecture. The THS1206, which is a 12-bit and 6-MSPS ADC, consists of a FIFO organized as a circular buffer to optimize the efficiency between ADC and DSP.

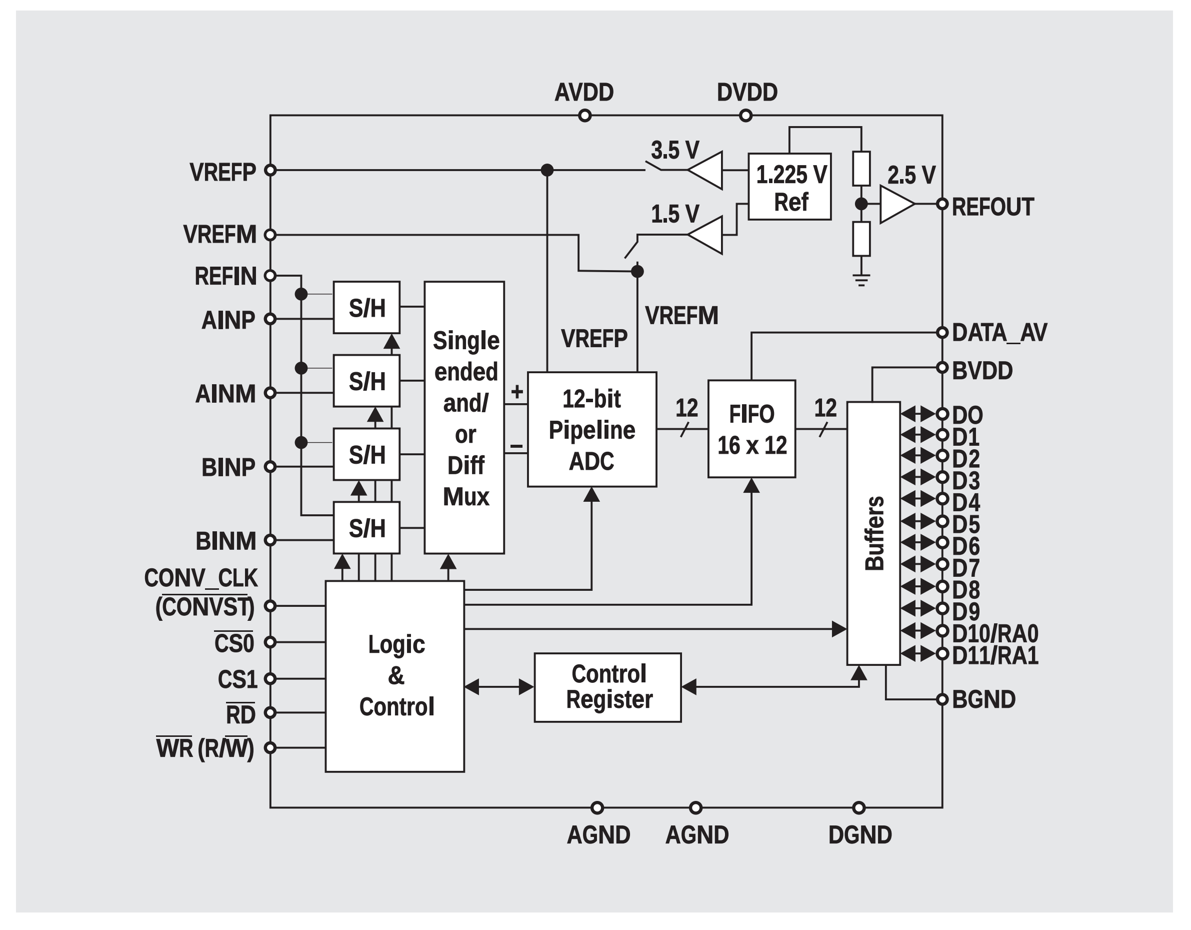
Figure 2 shows the block diagram of the THS1206. The main feature on the analog side is the provision of four analog input channels, which are switched simultaneously from the sample mode to the hold mode (simultaneous sample and hold). This feature is very important in modern control applications, radar and communications systems. The FIFO is capable of storing up to 16 conversion values and is located between the ADC core and the digital output of the THS1206.

Digital interface of the THS1206

When a digital interface for a DSP is evaluated, care should be taken to ensure that the THS1206 can be connected to different kinds of processors without external logic. To this end, one important criterion is the programmable reading and writing input. Processors today have either a combined reading/writing (R/$\overline{\text{W}}$) output or separate outputs for reading ($\overline{\text{RD}}$) and writing ($\overline{\text{WR}}$). The THS1206 can be connected to both types of processor without external logic. As can be seen in Figure 2, the THS1206 possesses a reading ($\overline{\text{RD}}$) input and a writing ($\overline{\text{WR}}$) input. The $\overline{\text{WR}}$ input can be reprogrammed on startup to a combined

Continued on next page

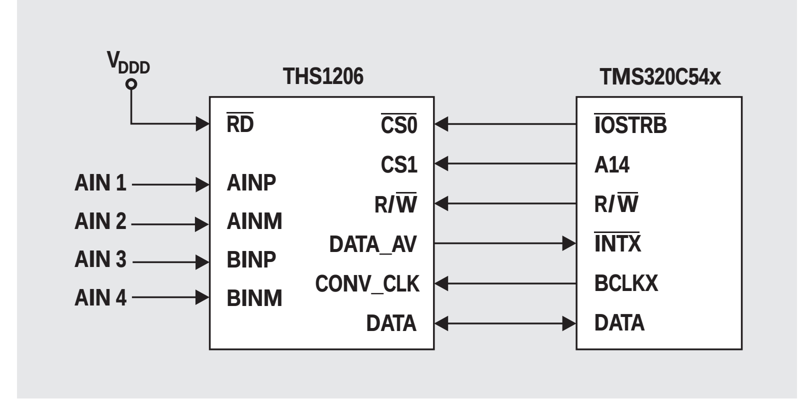
Figure 2. Block diagram of the THS1206



Continued from previous page

reading/writing input by a corresponding programming of the internal control register. With the examples of the two DSPs, TMS320C54x and TMS320C6201, the different possibilities are shown.

Figure 3. Interface of the THS1206 to the TMS320C54x



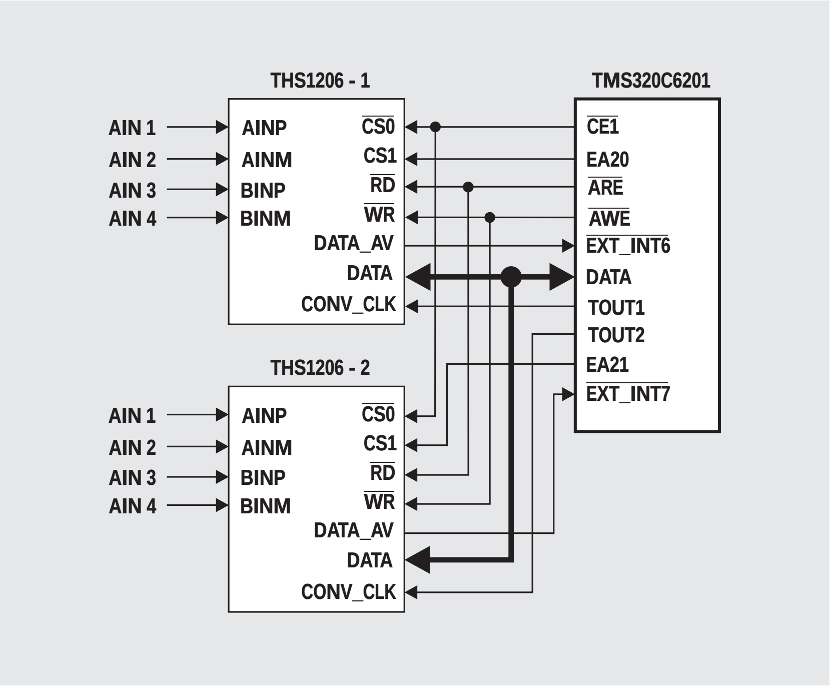
Interface of the THS1206 to the TMS320C54x

Figure 3 shows an example in which a THS1206 is connected to the DSP TMS320C54x. The TMS320C54x controls the reading and writing via the combined output R/$\overline{W}$. The R/$\overline{W}$ input of the THS1206 is programmed to this end as a combined reading and writing input. The $\overline{RD}$ input is switched to inactive in this operating mode. Addressing the THS1206 occurs via the two inputs $\overline{CS0}$ and CS1. The selection of the I/O peripherals is controlled via the output $\overline{IOSTRB}$ of the TMS320C54x, and addressing takes place via the address line A14. The converted data is read by TMS320C54x, controlled via an interrupt ($\overline{INTX}$).

Interface of the THS1206 to the TMS320C6201

Figure 4 shows an example in which two THS1206 ADCs are connected to the TMS320C6201 DSP. The TMS320C6201 controls the reading via the output $\overline{\text{ARE}}$ and the writing via the output $\overline{\text{AWE}}$. The selection of the THS1206 ADC is undertaken via the control output $\overline{\text{CE1}}$. The addressing of both ADCs occurs via the address connectors EA20 and EA21. An external address decoder therefore is not required. The converted data is at any time controlled via an interrupt (EXT_INT6, EXT_INT7) taken over by TMS320C6201.

Figure 4. Interface of the THS1206 to the TMS320C6201

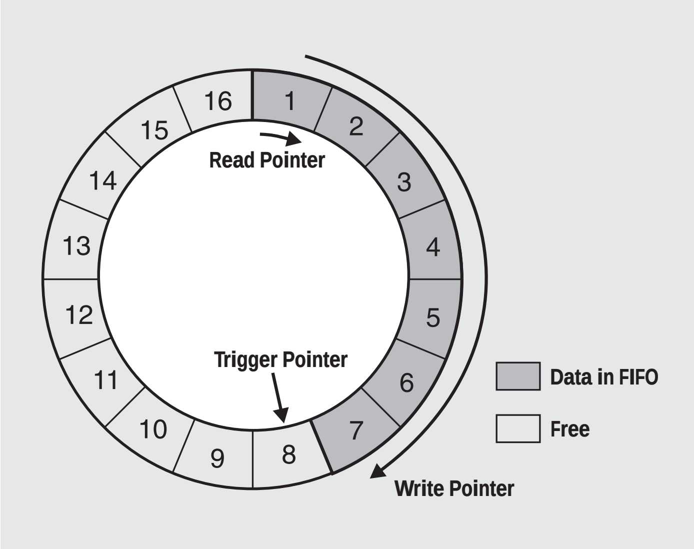


Integrated FIFO—circular buffer

As previously mentioned, the new THS1206 ADC has an on-chip FIFO, implemented as a flexible circular buffer to increase the maximum data throughput between ADC and DSP. With every falling edge of the sampling clock, a new conversion is initiated and a converted value is written into the FIFO. The circular buffer integrated in the THS1206 can store up to 16 conversion values. With a conversion rate of 6 MHz and a programmed FIFO depth of 8, the THS1206 stores 8 values in the FIFO within 8 x 167 ns = 1.34 µs. Only then does the ADC indicate to the DSP, with the digital control signal DATA_AV, that a block of data is ready to be read. This gives more flexibility to the processor and frees up resources for other activities.

The FIFO is configured as a flexible circular buffer. In principle it allows an overwriting of data that hasn't been read. This can be of interest for applications where data has to be sampled only in specific time frames. An example of this is the fast Fourier transformation (FFT), where typically a block of data has to be read (1024, 2048, 4096, ...).

Figure 5. FIFO organized as a circular buffer



However, the data that is read always should be the most recent, which is ensured by this specific circular buffer architecture. Overwriting is prevented in the relevant time frames by reading fast enough from the processor to prevent overlap.

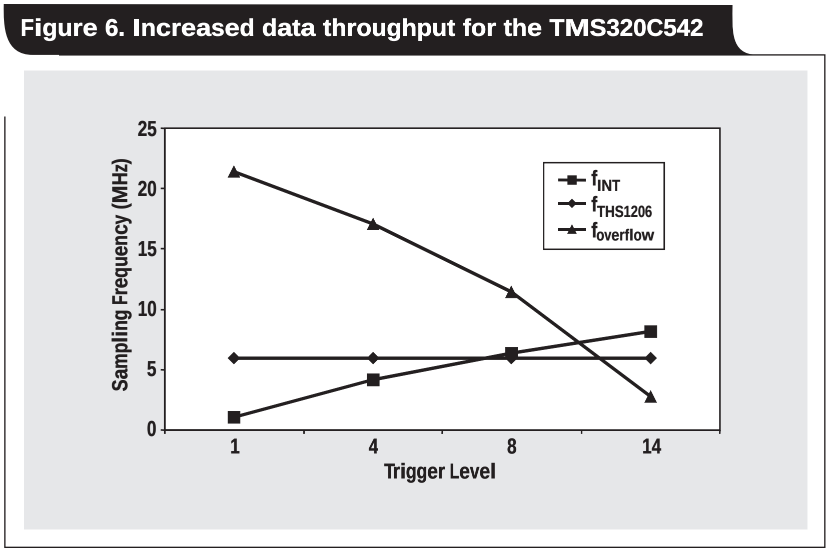
Figure 5 shows the architecture of the integrated FIFO, organized as a circular buffer. The reading from and writing to the FIFO is controlled entirely internally and can occur asynchronously. The conversion values are written automatically to the FIFO. In order to control the reading and writing, a read pointer and write pointer are used. These pointers always point to the cell in the FIFO where data has been written and to the cell that should be read next time.

This FIFO is designed specifically to allow multiple analog input channels for the ADC. In this case, the converted values are written into the FIFO in a predefined sequence (Autoscan Mode). Therefore, the channel information is always maintained for the processor and the higher data throughput can also be achieved for a data converter with multiple inputs.

Continued on next page

Continued from previous page

Figure 6. Increased data throughput for the TMS320C542



It is possible to select a specific storage depth (trigger level). When this level is reached, the converter (THS1206) informs the connected processor via the digital output DATA_AV (data available) that a block of converted values is ready to be transferred. The block size to be read is always equal to the setting of the storage depth (trigger level). A further pointer (trigger pointer) controls the signal DATA_AV dependent on the storage depth.

Figure 5 gives an example with a selected storage depth of 8. Initially, the trigger pointer points to position 8. The read pointer points to position 1, and the write pointer initially begins to write into location 1 and is incremented with every conversion value. In the example, the write pointer already points to position 7. The next increment of the write pointer reaches the trigger pointer, which means that the storage depth is reached (position 8). Therefore, the control signal DATA_AV becomes active and tells the processor that a block of data (8 values) waits to be read. The selectable storage depth allows variable settings for different processors and specific applications.

Efficiency of the FIFO

As previously explained, the conversion values of the ADC can be read in a block from the processor. This optimizes the interface between the ADC and the DSP, which frees up the processor and speeds up the data transfer.

To analyze the maximum data throughput between the processor and ADC, the latency time of the processor, the length of the ISR, and the storage depth of the FIFO are essential. The required parameters for an analysis of the maximum data throughput are:

T_F	Depth of the FIFOs
T_L	Trigger level
$t_{\Delta AVR}$	DATA_AV active to first read
t_{LAT}	DATA_AV active to first instruction within the ISR
t_{ISR}	Length of the ISR

The latency time is defined by the hardware and software of the DSP. The length of the interrupt service routine results from the latency time and the time required to read a block of data from the FIFO. The data throughput goes up with an increasing block size that can be read within one ISR. This is valid up to a specific limit of the FIFO, determined by the depth of the FIFO. As soon as the storage depth (trigger level) reaches the maximum depth of the FIFO, old data might be overwritten. Therefore, for every processor and specific application, the ideal choice of all parameters exists.

The maximum data throughput can be derived as follows. Equation 1 determines the maximum data throughput at a given trigger level (T_L) and depth of the FIFO (T_F) without overwriting of old data. Equation 2 determines the ability of the processor to capture the data fast enough at a given trigger level (T_L) and latency time (t_{LAT}).

$$f_{overflow} = \frac{T_F - T_L}{t_{\Delta AVR}} \tag{1}$$

$$f_{INT} = \frac{T_L}{t_{LAT} + t_{ISR}} \tag{2}$$

Figure 6 proves the efficiency of the FIFO. In this example, the THS1206 is connected to the TMS320C542 with a 40-MHz crystal. The figure shows the maximum sampling rate versus the selected storage depth (trigger level). The speed limitation caused by the length of the ISR (f_{INT}), the risk of overwriting unread data ($f_{overflow}$), and the maximum sampling speed of the ADC ($f_{THS1206}$) are shown. The highest data throughput at a specific trigger level is determined by the slowest of these three parameters. In this specific example, the maximum speed of the THS1206 can be achieved only if a trigger level of 8 is chosen for a 16-word-deep FIFO. At a trigger level below 8, the processor is unable to achieve the high data throughput because of the latency time. Overwriting of old data becomes critical with a trigger level higher than 8.

The maximum data throughput of the THS1206 into the TMS320C54x also is shown in Table 1.

Table 1. Data throughput of the TMS320C542-40

T_L	1	4	8	14
t_{ISR}	475 ns	550 ns	850 ns	1300 ns
f_{device}	6 MHz	6 MHz	6 MHz*	6 MHz
$f_{overflow}$	21.4 MHz	17.1 MHz	11.4 MHz	2.8 MHz*
f_{ISR}	1.1 MHz*	4.2 MHz*	6.4 MHz	8.2 MHz

*Maximum data throughput dependent on the selected trigger level

Figure 7 shows a second example of the FIFO's efficiency. In this case the THS1206 with integrated FIFO is connected to the TMS320C6701 with a 100-MHz crystal. The figure shows the maximum sampling rate versus the selected storage depth (trigger level). The speed limitation caused by the length of the ISR (f_{INT}), the risk of overwriting unread data ($f_{overflow}$), and the maximum sampling speed of the ADC ($f_{THS1206}$) are shown. The highest data throughput at a specific trigger level is determined by the slowest of these three parameters. In this specific example, the maximum speed of the THS1206 can be achieved if a trigger level higher than 1 is chosen for a 16-word-deep FIFO. Even a trigger level of 14 is possible in this example. The processor is fast enough to react and to prevent an overwriting.

The maximum data throughput of the THS1206 into the TMS320C6701 also is shown in Table 2.

The explanations and examples used in this article show that the way data converters and DSPs transfer data will have a significant effect on the overall performance of a system. System designers will find that the speed of these data transfers can be optimized when the data converters and DSPs have features and capabilities that complement each other. This requires an intimate knowledge of both DSP and data converter technology.

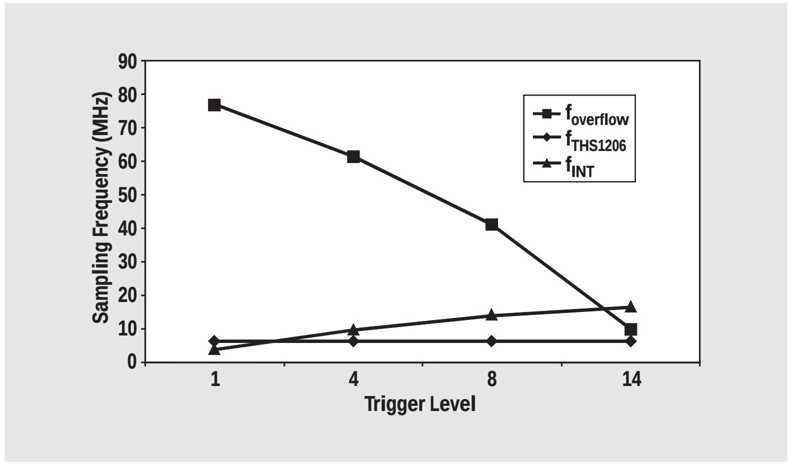
Texas Instruments provides a family of ADCs with FIFO. These are the THS10064 (10 bits, 6 MSPS, 4 analog inputs), THS12082 (12 bits, 8 MSPS, 2 analog inputs), and the THS10082 (10 bits, 8 MSPS, 2 analog inputs).

Table 2. Data throughput of the TMS320C6701-100

T_L	1	4	8	14
t_{ISR}	130 ns	280 ns	480 ns	780 ns
f_{device}	6 MHz	6 MHz*	6 MHz*	6 MHz*
$f_{overflow}$	76.9 MHz	61.5 MHz	41.0 MHz	10.2 MHz
f_{ISR}	3.9 MHz*	9.6 MHz	13.1 MHz	15.4 MHz

*Maximum data throughput dependent on the selected trigger level

Figure 7. Increased data throughput for the TMS320C6701



References

For more information related to this article, you can download an Acrobat Reader file at www-s.ti.com/sc/techlit/*litnumber* and replace *"litnumber"* with the **TI Lit. #** for the materials listed below.

Document Title	**TI Lit. #**
1. Data Book Data Acquisition Circuits (1998)	—
2. Data Converter Selection Guide (1998)	—
3. Mixed-Signal & Analog Products CD-ROM (1999)	—
4. Designing with the THS1206 High-Speed Data Converter	slaa094

Related Web sites

www.dataconverter.com
www.ti.com/sc/select
www.ti.com/sc/1206
www.ti.com/sc/docs/products/dsp/tms320c542.html
www.ti.com/sc/docs/products/dsp/tms320c54x.html
www.ti.com/sc/docs/products/dsp/tms320c6201.html
www.ti.com/sc/docs/products/analog/ths10064.html
www.ti.com/sc/docs/products/analog/ths12082.html
www.ti.com/sc/docs/products/analog/ths10082.html