ispPAC®20

In-System Programmable Analog Circuit

Features

- **IN-SYSTEM PROGRAMMABLE (ISP™) ANALOG**
  - **— Two Instrument Amplifier Gain/Attenuation Stages**
  - **— Signal Summation (Up to 3 Inputs)**
  - **— Precision Active Filtering (10kHz to 100kHz)**
  - **— 8-Bit DAC and Fast Dual Comparator**
  - **— Non-Volatile E²CMOS® Cells (10,000 Cycles)**
  - **— IEEE 1149.1 JTAG Serial Port Programming**
- **LINEAR ELEMENT BUILDING BLOCKS**
  - **— Programmable Gain Range (0dB to 40dB)**
  - **— Bandwidth of 550kHz (G=1), 330kHz (G=10)**
  - **— Low Distortion (THD < -74dB max @ 10kHz)**
  - **— Auto-Calibrated Input Offset Voltage**
- **TRUE DIFFERENTIAL I/O**
  - **— High CMR (69dB) Instrument Amplifier Inputs**
  - **— 2.5V Common Mode Reference on Chip**
  - **— Rail-to-Rail Voltage Outputs**
  - **— Single Supply 5V Operation**
- **44-PIN PLASTIC PLCC AND TQFP PACKAGES**
- **APPLICATIONS INCLUDE INTEGRATED:**
  - **— Single +5V Supply Signal Conditioning**
  - **— Active Filters, Gain Stages, Summing Blocks**
  - **— Analog Front Ends, 12-Bit Data Acq. Systems**
  - **— Precision Voltage Controlled Oscillator**
  - **— Synchronous Detection Circuits**
  - **— Precision Rectification & Other Non-Linear Functions**

Description

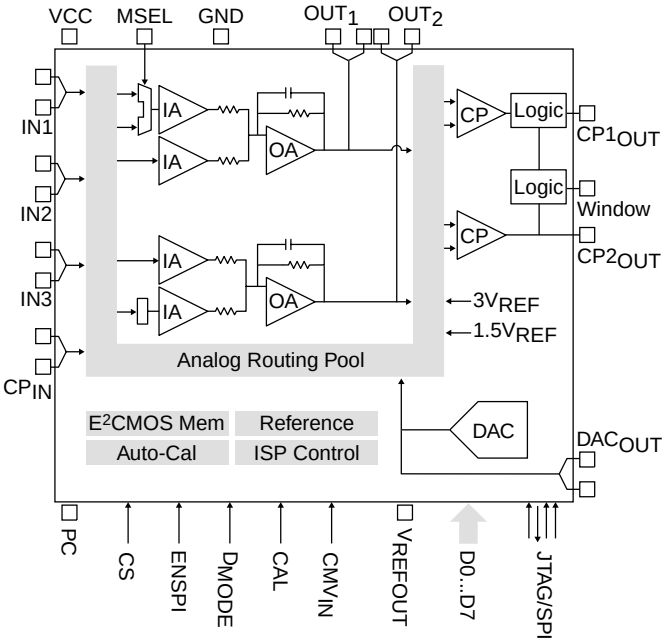
The ispPAC20 is a member of the Lattice family of In-System Programmable analog circuits, digitally configured via nonvolatile E²CMOS technology.

Analog building blocks, called PACblocks, replace traditional analog components such as opamps and active filters, eliminating the need for most external resistors and capacitors. Also included are an 8-bit DAC and dual comparators. With no requirement for external configuration components, ispPAC20 expedites the design process, simplifying prototype circuit implementation and change, while providing high-performance integrated functionality.

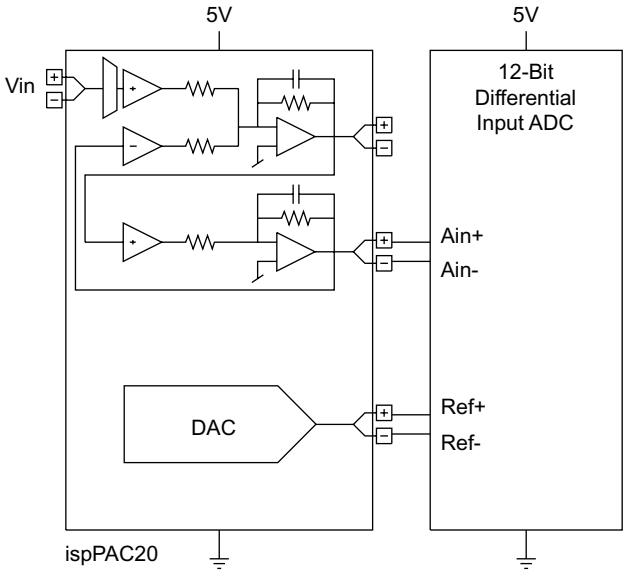
Designers configure the ispPAC20 and verify its performance using PAC-Designer®, an easy-to-use, Microsoft Windows® compatible program. Device programming is supported using PC parallel port I/O operations.

The ispPAC20 is configured through its IEEE Standard 1149.1 (JTAG) compliant serial port. The flexible In-System Programming capability enables programming, verification and reconfiguration if desired, directly on the printed circuit board.

Functional Block Diagram



Typical Application Diagram



LATTICE SEMICONDUCTOR CORP., 5555 Northeast Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. (503) 268-8000; 1-800-LATTICE; FAX (503) 268-8556; http://www.latticesemi.com

May 2001

T_A = 25°C; V_S = 5.0V; Signal path = V_{IN} to V_{OUT} of one PACblock (second input unused); 1V ≤ V_{OUT} ≤ 4V; Gain = 1; Output load = 200pf, 1MΩ. Feedback enabled; Feedback capacitor = minimum; Auto-cal initiated immediately prior. (Unless otherwise specified).

DC Electrical Characteristics

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNITS
PACblock Analog Input						
$V_{IN\pm}$ (1)	Input Voltage Range	Applied to Either V_{IN+} or V_{IN-}	1		4	V
$V_{IN-DIFF}$	Differential Input Voltage Swing (2)	2| $V_{IN+} - V_{IN-}$ |	6			V_{p-p}
V_{OS} (2)	Differential Offset Voltage (Input Referred)	G = 10		20	100	µV
		G = 1		0.2	1.0	mV
$\Delta V_{OS}/\Delta T$	Differential Offset Voltage Drift	-40 to +85°C		50		µV/°C
R_{IN}	Input Resistance			10^9		Ω
C_{IN}	Input Capacitance			2		pF
I_B	Input Bias Current	at DC		3		pA
e_N	Input Noise Voltage Density	At 10kHz, Referred to Input, G = 10		38		nV/√Hz
PACblock Analog Output						
$V_{OUT\pm}$	Output Voltage Range	Present at Either V_{OUT+} or V_{OUT-}	0.1		4.9	V
$V_{OUT-DIFF}$	Differential Output Voltage Swing (2)	2| $V_{OUT+} - V_{OUT-}$ |	9.6			V_{p-p}
$I_{OUT\pm}$	Output Current	Source/Sink	10			mA
V_{CM}	Common Mode Output Voltage	$(V_{OUT+} + V_{OUT-})/2$; $V_{IN+} = V_{IN-}$	2.475	2.500	2.525	V
PACblock Static Performance						
G	Programmable Gain Range	Each individual PACblock	0		26	dB
	Gain Error	R_L = 300Ω Differential			4.0	%
	Gain Matching	Between Two Inputs of Same PACblock			3.0	%
$\Delta G/\Delta T$	Gain Drift	-40 to +85°C		20		ppm/°C
PSR	Power Supply Rejection	Differential at 1kHz		80		dB
		Single-ended at 1kHz		77		dB
Common Mode Reference Output ($VREF_{OUT}$)						
$VREF_{OUT}$	Output Voltage Range	Nominally 2.500V	-0.2		+0.2	%
CMV_{IN} (4)	Common Mode Output Voltage Input	Optional External $VREF_{OUT}$ Reference Voltage	1.25		3.25	V
	Output Voltage Drift	-40 to +85°C		50		ppm/°C
$IREF_{OUT}$	Output Current	Source		50		µA
		Sink		350		µA
	Output Noise Voltage	10MHz Bandwidth; 1µF Bypass Capacitor		40		μV_{RMS}
	Power Supply Rejection	1kHz		80		dB
Digital-to-Analog Converter (DAC) PACell						
	Resolution		8			bits
INL	Integral Non-Linearity Error				±0.5	lsb
DNL	Differential Non-Linearity	Guaranteed Monotonic			±1.0	lsb
	Gain Error				2.5	%
$\Delta/\Delta T$	Gain Drift	-40 to +85°C		20		ppm/°C
V_{OS}	Differential Offset Voltage				2	mV
V_{CM}	Common Mode Output Voltage	$(D_{OUT+} + D_{OUT-})/2$	2.495	2.500	2.505	V
PSR	Power Supply Rejection	Differential at 1kHz		80		dB
$\Delta V_{OS}/\Delta T$	Differential Offset Voltage Drift	-40 to +85°C		50		µV/°C
FSR	Differential Full Scale Range	DAC Code 00h to FFh		6.0		V
$V_{OUT\pm}$	Voltage Output Range	R_L = 1KΩ Differential	1		4	V
$I_{OUT\pm}$	Output Current	Source/Sink	10			mA
SR	Output Slew Rate			1.3		V/µs
t_S	Output Settling Time 0.1%	$6V_{DIFF}$ Input Step		4.8	6.0	µs
Temperature Range						
	Operation		-40		+85	°C
	Storage		-65		+150	°C

DC Electrical Characteristics (Continued)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNITS
Comparator PACells						
A_V	Voltage Gain			108		dB
V_{OS}	Input Offset Voltage			5		mV
$\Delta V_{OS}/\Delta T$	Differential Offset Voltage Drift	-40 to +85°C		50		µV/°C
PSR	Power Supply Rejection	Differential at 1kHz		80		dB
	Programmable Hysteresis	On or Off		±47		mV
t_P	Propagation Delay	Overdrive = 10mV		750		ns
		Overdrive = 100mV		150		ns
	Input Common Mode Input Range		0		5.0	V
CMRR	Input Common Mode Rejection Ratio			60		dB
Programming						
	Erase Program Cycles		10K			cycles
Digital I/O						
V_{IL}	Input Low Voltage		0		0.8	V
V_{IH}	Input High Voltage		2.0		V_S	V
I_{IL}, I_{IH}	Input Leakage Current	0V ≤ TCK Input ≤ V_S			±10	µA
		0V ≤ All Other Inputs ≤ V_S			+40/-70	µA
V_{OL} (5)	Output Low Voltage	I_{OL} = 4.0mA			0.5	V
V_{OH} (5)	Output High Voltage	I_{OH} = -1.0mA	2.4			V
Power Supplies						
V_S	Operating Supply Voltage		4.75	5.0	5.25	V
I_S	Supply Current	V_S = 5.0V			21	mA
P_D	Power Dissipation	V_S = 5.0V			105	mW

AC Electrical Characteristics

SYMBOL	PARAMETER		CONDITION	MIN.	TYP.	MAX.	UNITS
PACblock Dynamic Performance							
THD	Total Harmonic Distortion	Differential	F_{IN} = 10kHz		-88	-74	dB
		Single-Ended			-72		dB
		Differential	F_{IN} = 100kHz		-67	-62	dB
		Single-Ended			-63		dB
SNR	Signal to Noise	G = 1 to 10	0.1Hz to 100kHz		103		dB
CMR	Common Mode Rejection (V_{IN} = 1V to 4V)		10kHz		69		dB
	Note: V_{IN+} and V_{IN-} connected together		100kHz		55		dB
BW	Small Signal Bandwidth	G = 1			550		kHz
		G = 10			330		kHz
BW_{FP}	Full Power Bandwidth		V_{IN} = 6V_{DIFF}, V_{OUT} = -3dB, G=1		330		kHz
SR	Slew Rate			5.0	7.5		V/µs
t_S	Settling Time	0.1%	6V_{DIFF} Input Step		2.0		µs
	Crosstalk		Between Any Two Channels		-90		dB
PACell Filter Characteristics							
	Filter Pole Programming Range		Number of Poles in Range > 120	10		100	kHz
F_0	Absolute Pole Frequency Accuracy		Deviation From Calculated Value		1.0	5.0	%
ΔF_0	Pole Step Size (Between Calculated Poles)		10kHz to 100kHz			3.2	%
DF_0/DT	Pole Frequency Change vs. Temperature		-40 to +85°C		0.02		%/°C

Notes: (1) A wider input range of 0.7V to 4.3V is typical, but not guaranteed. Inputs larger than this will be clipped. Input signals are also subject to common-mode voltage limitations. Refer to the table of conditions in this datasheet. (2) Refer to theory of operation section later in this datasheet for explanation of differential voltage swing computation. (3) To insure full spec performance an additional auto-calibration should be performed after initial turn-on and the device reaches thermal stability.(4) The user-provided voltage on this pin (CMV_{IN}) becomes an optional (selected via programming) alternative to the default 2.5V $VREF_{OUT}$. (5) Includes TDO, CP1OUT, CP2OUT and WINDOW output logic pins.

Absolute Maximum Ratings

Supply Voltage V_S ..... -0.5 to +7V
Logic and Analog Input Voltage Applied ..... 0 to V_S
Logic and Analog Output Short Circuit Duration ..... Indefinite
Lead Temperature (Soldering, 10 sec.) ..... 260°C
Ambient Temperature with Power Applied ... -55 to 125°C
Storage Temperature ..... -65 to 150°C

Note: Stresses above those listed may cause permanent damage to the device. These are stress only ratings and functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied.

ispPAC20 Ordering Information

Ordering Number	Package
ispPAC20-01JI	44-Pin PLCC
ispPAC20-01TI	44-Pin TQFP

Package Options

44-Pin PLCC

44-Pin TQFP

Part Number Description

ispPAC 20 – XX X X

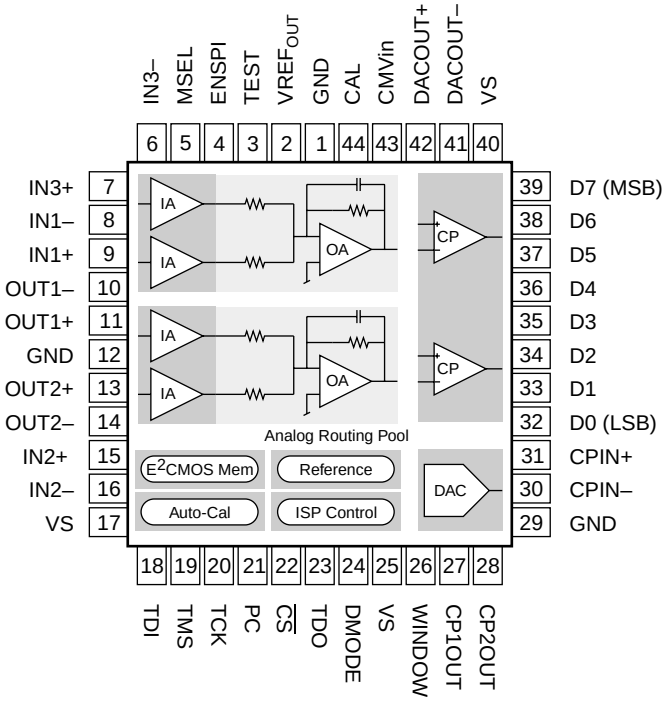
Device Family

Device Number

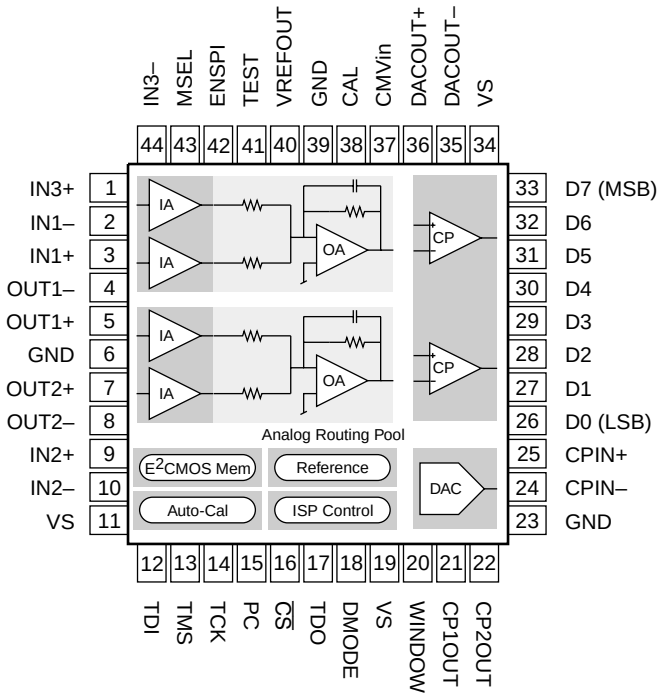
Performance Grade
01 = Standard

Package
J = PLCC
T = TQFP

Grade
I = Industrial Temperature



Pin Diagram
44 PLCC Package

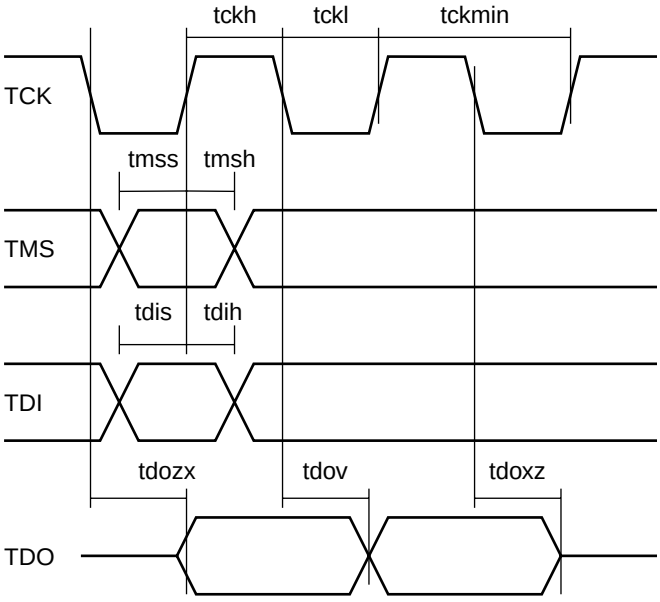


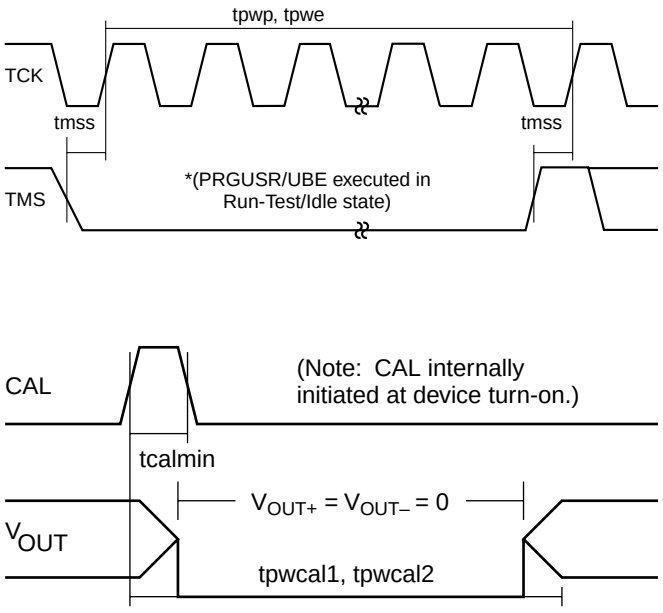
Pin Diagram
44 TQFP Package

Timing Specifications (JTAG Interface Mode)

T_A = 25°C; V_S = +5.0V (Unless otherwise specified)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNITS
Dynamic Performance						
tckmin	Minimum Clock Period		200			ns
tckh	TCK High Time		50			ns
tckl	TCK Low Time		50			ns
tmss	TMS Setup Time		15			ns
tmsh	TMS Hold Time		10			ns
tdis	TDI Setup Time		15			ns
tdih	TDI Hold Time		10			ns
tdozx	TDO Float to Valid Delay				60	ns
tdov	TDO Valid Delay				60	ns
tdoxz	TDO Valid to Float Delay				60	ns
tpwp	Time for a programming operation	Executed in Run-Test/Idle	80		100	ms
tpwe	Time for an erase operation	Executed in Run-Test/Idle	80		100	ms
tpwcal1	Time for auto-cal operation on power-up	Automatically executed at power-up			250	ms
tcalmin	Minimum auto-cal pulse width		40			ns
tpwcal2	Time for user initiated auto-cal operation	Executed on rising edge of CAL			100	ms





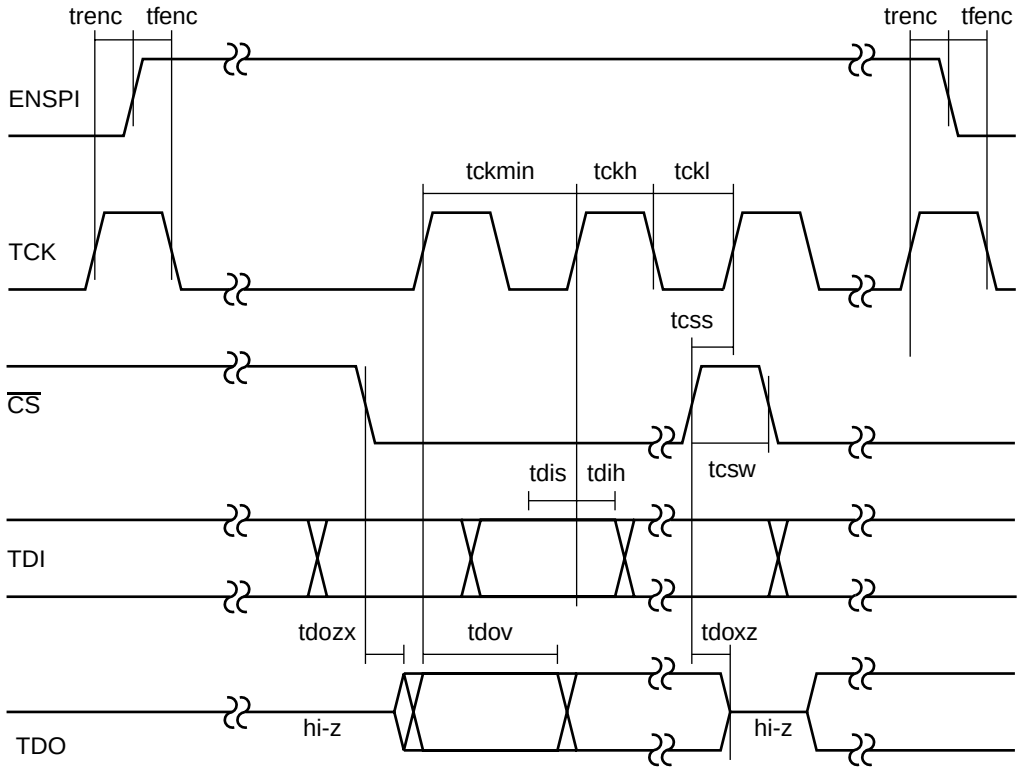
*Note: During device JTAG programming, filsum PACblock analog outputs will stop responding to normal input stimulus. This is because all configuration information is erased and then re-written as part of a normal programming cycle, momentarily disrupting the input to output signal path. Behavior is not predictable during either of these steps since the analog outputs are not clamped during a programming cycle. Usually, however, the outputs will slew to either 0V (Ground) or 5V (V_{supply}) or 2.5V ($VREF_{OUT}$). This behavior is partially determined by conditions existing immediately prior to device reprogramming and intermediate configurations that occur during the process. DAC outputs will go to -FS (-3V_{DIFF}) during bulk erase and then to +FS (+3V_{DIFF}) for less than 2ms during final programming before assuming the programmed code value. Comparator outputs can change due to a number of additional factors and are therefore not predictable until the final device configuration is reached. Also, any configuration of the comparators that modifies their mode of operation (e.g., hysteresis on, clocked output mode, etc) can alter output states from initial settings until additional external conditions are reapplied to the device.

Timing Specifications (SPI/Parallel Interface Modes)

T_A = 25°C; V_S = +5.0V (Unless otherwise specified).

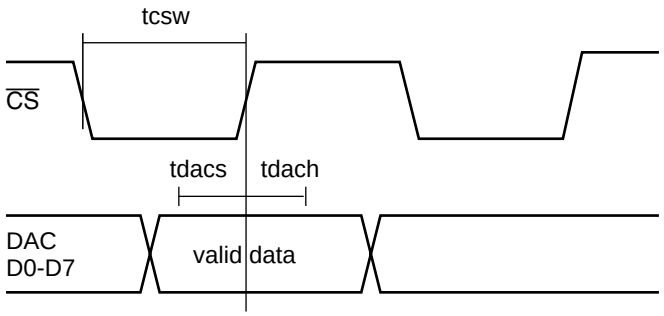
SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNITS
Dynamic Performance						
trenc	Minimum Rising Clock to ENSPI Time		10			ns
tfenc	Minimum ENSPI to Falling Clock Time		10			ns
tckmin	Minimum Clock Period		100			ns
tckh	TCK High Time		50			ns
tckl	TCK Low Time		50			ns
tcss	CS Setup Time		35			ns
tcsw	Minimum CS Pulse Widths		40			ns
tdis	TDI Setup Time		15			ns
tdih	TDI Hold Time		10			ns
tdacs	DAC Data Setup Time		15			ns
tdach	DAC Data Hold Time		10			ns
tdozx	TDO Float to Valid Delay				60	ns
tdov	TDO Valid Delay				60	ns
tdoxz	TDO Valid to Float Delay				60	ns

Timing Specifications (SPI Interface Mode)

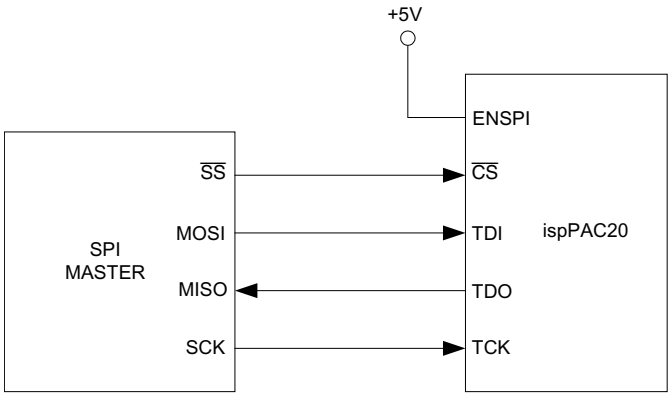


Timing Specifications (SPI/Parallel Interface Modes), Continued

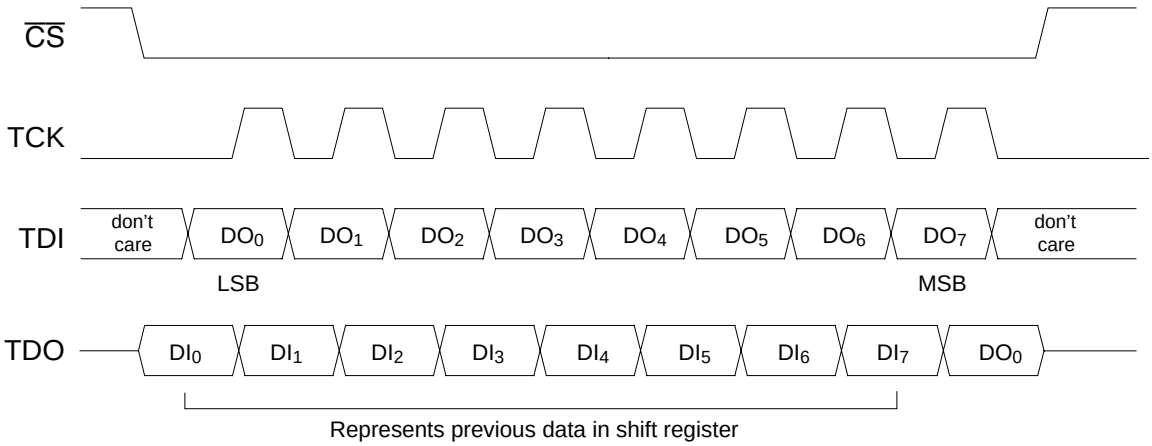
DAC Parallel Input Timing Specifications



SPI Connection Diagram



SPI Data Transfer



Notes

1. SPI data is loaded in TDI, LSB first. If TCK continues to clock after $\overline{CS}$ goes high, data will continue to be shifted through the shift register, even though the TDO pin is tristated after $\overline{CS}$ goes high.
2. DO_0 –> DO_7 represents "data out" from the SPI microprocessor or other digital source to the TDI input of the ispPAC20.
3. DI_0 –> DI_7 represents "data in" from the ispPAC20 TDO pin to the SPI microprocessor input or other digital source.
4. After the eighth clock, the LSB (DO_0) is valid on TDO as long as $\overline{CS}$ is low.

Pin Descriptions

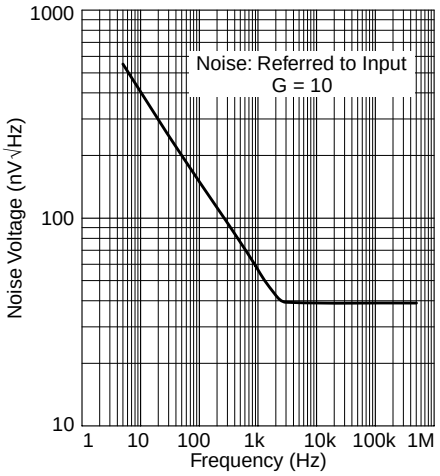
Pin(s)				
TQFP	**PLCC**	**Symbol**	**Name**	**Description**
39, 6, 23	1,12,29	GND	Ground	Ground pins. All should normally be connected to same analog ground plane.
40	2	VREFout	Common-Mode Reference	Common-mode voltage reference output pin (+2.5V nominal). Must be bypassed to GND with a 1µF capacitor.
43	5	MSEL	Multiplexer Control	Multiplexer logic input pin. Selects either of two analog channels to one of the PACblock inputs. Input A selected when low, B when high. Internal pull-down to GND.
42	4	ENSPI	Enable SPI Mode	Enable SPI logic input pin. When high, causes serial port to run in SPI mode. Internal pull-down to GND.
41	3	TEST	Factory Test pin	Factory Test pin. Connect to GND for proper circuit operation.
44, 1, 2, 3, 9, 10	6, 7, 8, 9, 15, 16	IN	Inputs 1, 2, 3 (+ or -) Plus or minus	Differential input pins, with two pins per input (e.g., IN2+ and IN2-). components of V_{IN}, where differential $V_{IN} = V_{IN+} - V_{IN-}$.
4, 5, 7, 8	10, 11, 13, 14	OUT	Outputs 1,2 (+ or -)	Differential output pins, with two pins per output (e.g., OUT2+ and OUT2-). Complementary with respect to $VREF_{OUT}$, where differential $V_{OUT} = V_{OUT+} - V_{OUT-}$.
11, 19, 34	17, 25, 40	VS	Supply Voltage	Analog supply voltage pins (5V nominal). Must all be connected together. Should all be bypassed to GND with 1µF and .01µF capacitors.
12	18	TDI	Test Data In	Serial interface logic pin (input) for both JTAG and SPI modes. Input data valid on rising edge of TCK (JTAG). Internal pull-up to V_S.
13	19	TMS	Test Mode Select	Serial interface logic mode select pin (input). JTAG interface mode only. Internal pull-up to V_S.
14	20	TCK	Test Clock	Serial interface logic clock pin (input).
15	21	PC	Polarity Control	Polarity logic input pin. Controls polarity of one PACblock input. Operation determined by user configuration of device. Internal pull-down to GND.
16	22	$\overline{CS}$	Chip Select	Chip select logic input pin. SPI data and DAC parallel interface clock. Internal pull-up to V_S.
17	23	TDO	Test Data Out	Serial interface logic pin (output) for both JTAG and SPI operation modes. Output data valid on falling edge of TCK (JTAG).
18	24	DMODE	DAC Mode Select	DAC mode logic input. When high, DAC can be loaded via the parallel interface pins D0-D7 using CS as the latch command. Internal pull-down to GND.
20	26	WINDOW	Window	Window comparison logic pin (output). Configured by user to Comparator Outperform comparator logic functions.
21, 22	27, 28	CPOUT	Comparator Outputs	Comparator logic pins (outputs). One pin for logic level of each comparator.
24, 25	30, 31	CPIN	Comparator Inputs	Differential input pins, CPIN+ and CPIN-. Plus and minus components of V_{IN}, where differential $CP_{IN} = CP_{IN+} - CP_{IN-}$.
26 to 33	32 to 39	D0 to D7	DAC Data Inputs	DAC data pins (inputs). Eight parallel inputs to DAC. Clocked by CS pin. D0 is the LSB and D7 is the MSB.
35, 36	41, 42	DACOUT	DAC Outputs (+ or -)	Differential output pins (DOUT+ and DOUT-). Complementary with respect to VREFout, where differential $D_{OUT} = D_{OUT+} - D_{OUT-}$.
37	43	CMVin	Input for Optional $VREF_{OUT}$	Input pin for optional analog Common Mode Output Voltage (CMVin). Replaces VREFout (+2.5V) with this voltage for any user-selected PACblock.
38	44	CAL	Auto-Calibrate	Digital pin (input). Commands an auto-calibration sequence on a rising edge. Internal pull-down to GND.

Connection Notes

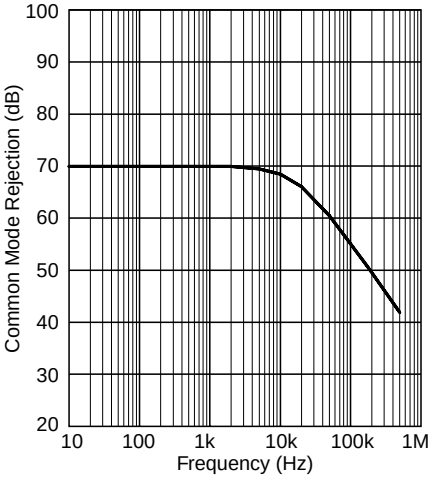
1. All inputs and outputs are labeled with plus (+) and minus (-) signs. Polarity is labeled for reference and can be selected externally by reversing pin connections or internally under user programmable control.
2. All analog output pins are “hard-wired” to internal output devices and should be left open if not used. Outputs of uncommitted PACblocks are forced to $VREF_{OUT}$ (2.5V) and can be used as low impedance reference output buffers. V_{OUT+} and V_{OUT-} should not be tied together as unnecessary power will be dissipated.
3. When the signal input is single-ended, the other half of the unused differential input must be connected to a DC common-mode reference (usually $VREF_{OUT}$, 2.5V).

Typical Performance Characteristics

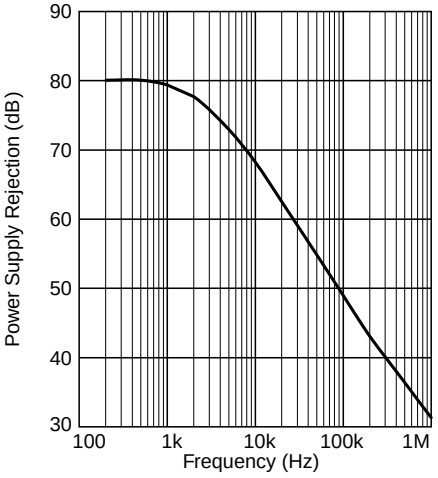
Input Noise Spectrum



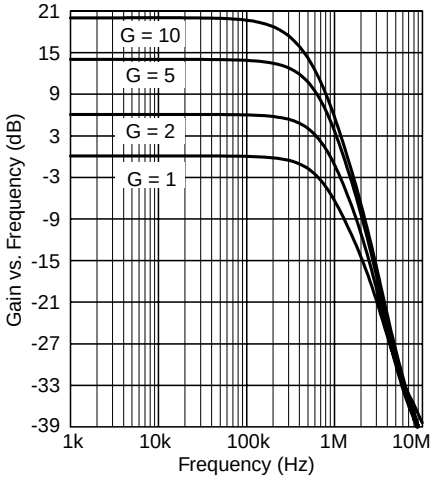
CMR vs. Frequency



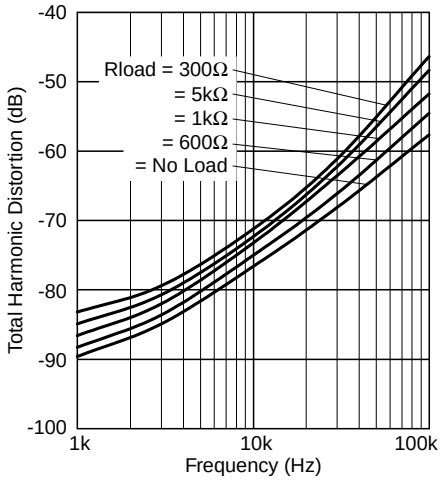
PSR vs. Frequency



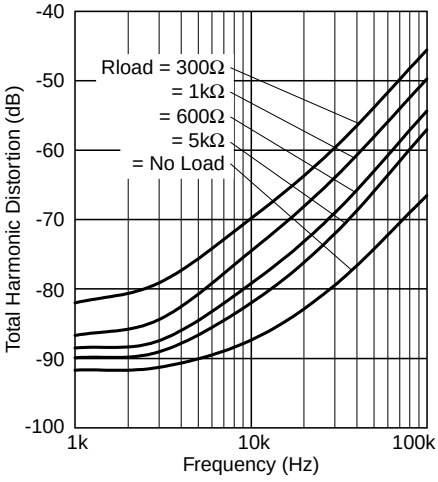
Small Signal BW vs. Gain



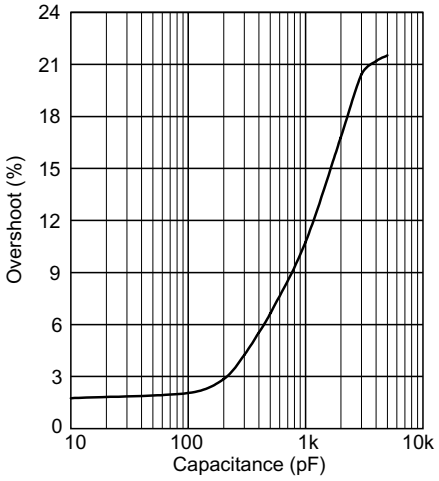
THD vs. Frequency (Gain=1)



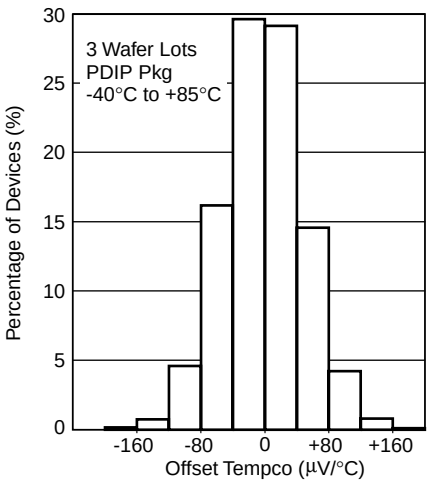
THD vs. Frequency (Gain=10)



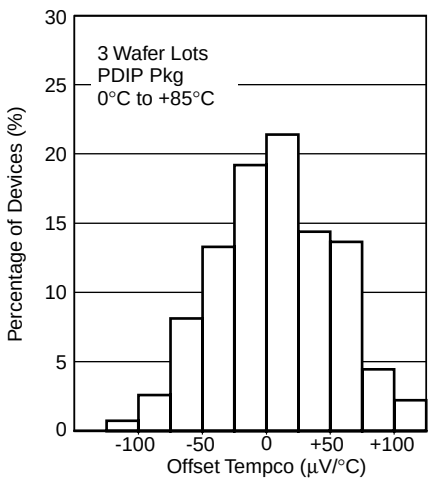
Capacitive Load Handling



V_{OS} Tempco

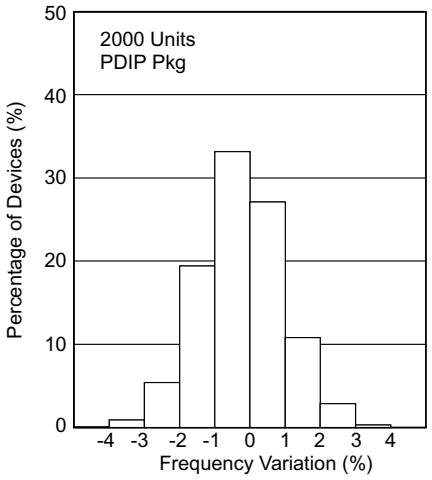


$VREF_{OUT}$ Tempco

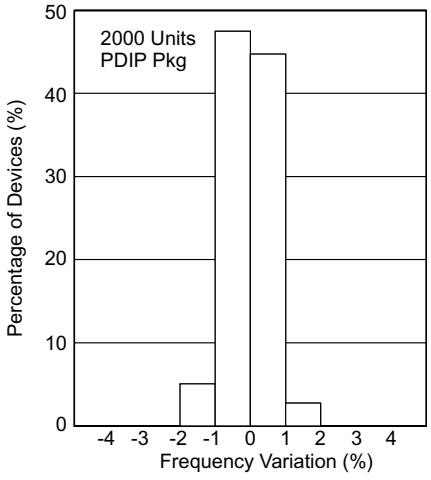


Typical Performance Characteristics

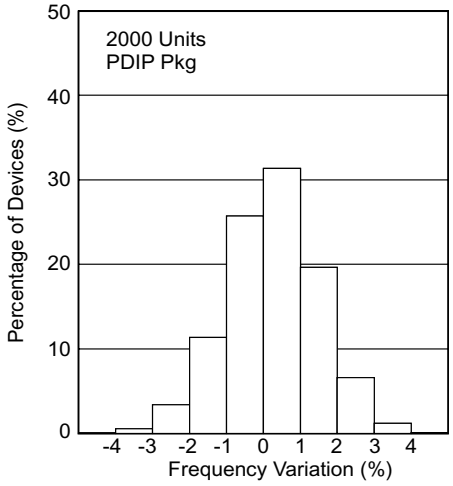
10.34kHz Filter F_c Accuracy



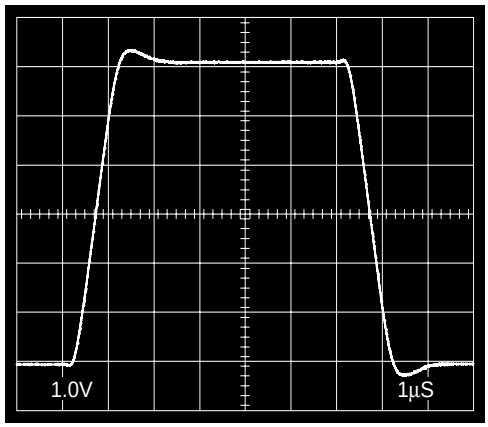
46.46kHz Filter F_c Accuracy



91.98kHz Filter F_c Accuracy

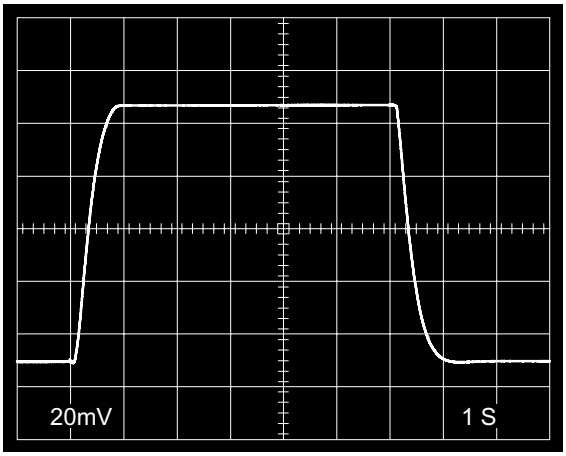


Large-Signal Response



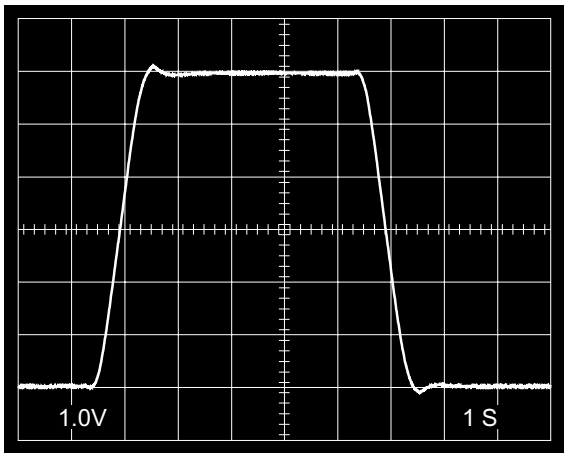
Gain = 1
Load = No Load

Small-Signal Response



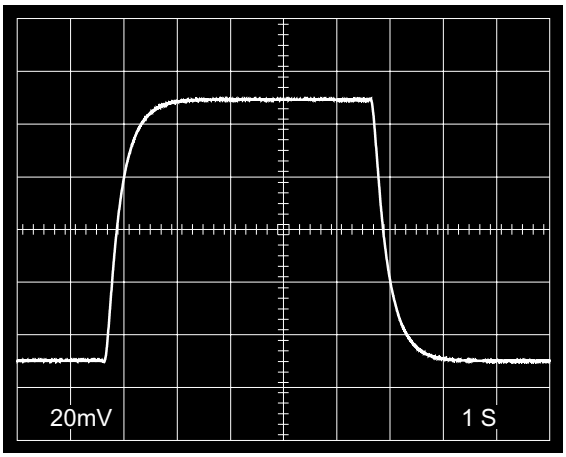
Gain = 1
Load = No Load

Large-Signal Response with 600pF Load



Gain = 1
Load = 600pF

Small-Signal Response with 600pF Load



Gain = 1
Load = 600pF

Theory of Operation

Introduction

The ispPAC20 includes two programmable analog macrocells called PACblocks, each emulating a collection of operational amplifiers, resistors and capacitors. Requiring no external components, it flexibly implements basic analog functions such as precision filtering, summing/differencing, gain/attenuation and integration. Each PACblock contains a summing amplifier, two differential input instrument amplifiers, and an array of feedback capacitors. The capacitors, combined with a fixed value feedback element, provide more than 120 programmable poles between 10kHz to 100kHz with an absolute accuracy of 5.0 percent. Variable gain input instrument amplifiers make it possible to program any PACblock gain in integer steps between ±1 and ±10. More complex signal processing functions are performed by configuring both PACblocks in combination with each other to achieve a variety of circuit functions.

The ispPAC20 architecture is fully differential from input to output. This effectively doubles dynamic range versus single-ended I/O. It also affords improved performance with regard to specifications such as input common mode rejection (CMR) and total harmonic distortion (THD).

Differential peak-peak voltage is determined by knowing the signal extremes on both differential input or output pins. For example, if V(+) equals 4V and V(-) equals 1V, the differential voltage is defined as V(+) - V(-) = Vdiff, or 4V - 1V = +3V. Since either polarity can exist on differential I/O pins, it is also possible for the opposite extreme to exist and would mean when V(+) equals 1V and V(-) equals 4V, the differential voltage is now 1V - 4V = -3V. To calculate the differential peak-peak voltage or full signal swing, the absolute difference between the two extreme Vdiff's is calculated. Using the previous examples would result in |(+3V) - (-3V)| = 6V. It can be immediately seen that true differential signals result in a doubling of usable dynamic range. For more explanation of this and other differential circuit benefits, please refer to application note AN6019.

Input polarity is programmable without affecting input impedance or dynamic performance, since no internal change is made other than routing to the input amplifier. Single-ended operation is achieved by using either one input and/or one output pin, as required, and adjusting gain settings to achieve desired output levels.

The ispPAC20 operates on a single 5V supply and includes an internal reference generating 2.5V. This reference is made available externally through the voltage common-mode reference or $VREF_{OUT}$ pin. The output common mode voltage is always referenced to 2.5V, regardless of the input common mode level. It is possible, when desired, to use an externally supplied voltage instead of $VREF_{OUT}$, however. This optional common-mode output voltage (V_{CM}) must be provided by the user via the CMV_{IN} input pin. The only limitation is this reference voltage must be between 1.25V and 3.25V. When an external voltage is present, an ispPAC20 must be programmed, on a per-PACblock basis, to use the external reference instead of the internal 2.5V.

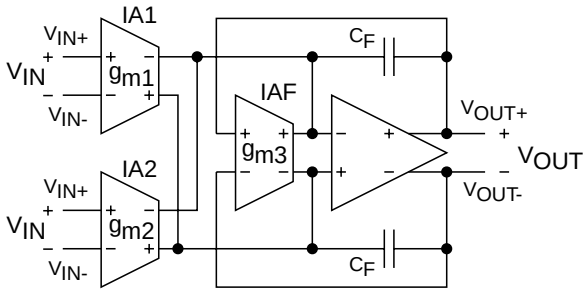
Configuring an ispPAC20 is accomplished using PAC-Designer, a Windows-based design environment. PAC-Designer includes an AC simulator for design verification prior to programming. The user can download the design to the ispPAC20 at any time via the device's IEEE Standard 1149.1 (JTAG) compliant serial port directly from the parallel port of a PC using an ispDOWNLOAD™ cable. Once downloaded, the circuit topology and component values are stored in non-volatile digital E^2CMOS cells on the ispPAC20 without any need for external programming voltages.

Architecture

In all ispPAC products, individual programmable circuit functions called PACells™ are carefully combined to form larger analog macrocells or PACblocks. The ispPAC20 has two such PACblocks that incorporate specially configured PACells to perform amplification, summation, integration and filtering. Each of the two filtering/summation or "FilSum" PACblocks within ispPAC20 is comprised of three separate PACells, two input instrument amplifiers and an output summing amplifier (see Figure 1). The input amplifier PACells act as front-end gain stages for the FilSum PACblock and allow multiple signals to be summed together. The PACblock's output amplifier is similar to the familiar operational amplifier except that it has true differential outputs. Also included with each output amplifier is a filter capacitor array and switchable DC feedback path element. These components in combination enable the filtering and integrating functions of the FilSum PACblock.

Theory of Operation (Continued)

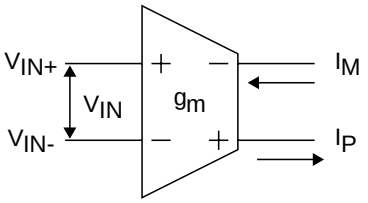
Figure 1. FilSum (Filtering/Summation) PACblock Diagram



Each FilSum PACblock actually employs three instrument amplifier (IA) PACells: two at the input (IA1 and IA2) and one as a feedback element around the op amp (IAF). The instrument amplifier PACells all have differential I/O and convert an input voltage to an output current (refer to Figure 2). This type of amplifier is sometimes referred to as an operational transconductance amplifier or OTA. When a differential input voltage is applied to these IAs, it is converted to a current proportional to the input signal.

Because an AC signal common to both of the high impedance inputs of the IA does not create a net difference in the input signal, it is rejected by the amplifier. This characterizes the function of what is commonly known as an instrument amplifier and is a very desirable property because it acts to preserve the integrity of small signals in the presence of otherwise overwhelming noise.

Figure 2. Instrument Amplifier PACell



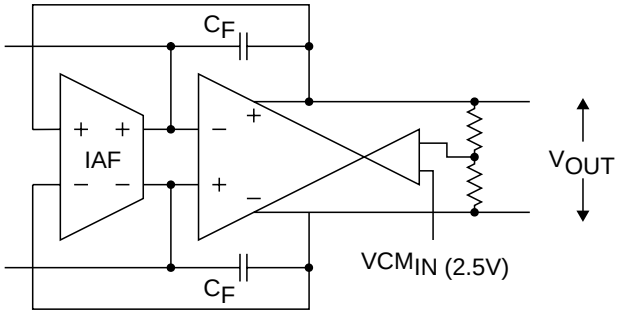
The two input instrument amplifiers have a programmable transconductance (g_m) value in 10 steps between 2µA/V and 20µA/V with programmable input polarity, whereas the feedback amplifier is fixed at 2µA/V. The IA PACells exhibit extremely high input impedance so they don't load circuitry driving them and their outputs can be enabled or disabled under E^2CMOS control, effectively switching them in and out of the FilSum PACblock circuitry. These simple characteristics permit a great deal of functionality: Signals can be summed, the resistive amplifier feedback can be removed to create an integrator, the sign of PACblock transfer function can be changed without changing the input or output loading characteristics. The FilSum PACblock can precisely filter, amplify or attenuate signals, always maintaining the high impedance input qualities of instrumentation amplifiers.

FilSum PACblock Operation

All ispPAC20 inputs are differential, the input signal being the difference between input amplifier (IA) PACell pins V_{IN+} (Positive Input) and V_{IN-} (Minus Input). The common mode value of the input is ignored, and as long as the inputs are not within one volt of the supply rails, the part is in its linear operating region. As the input signal range exceeds these limits, distortion begins to increase until clipping occurs. This is discussed further in the advanced topics section.

The output is also differential, being the difference between output amplifier (OA) PACell pins V_{OUT+} and V_{OUT-}. The output maintains high linearity to within 100mV of the supply rails under minimum load. The output has short circuit protection and is capable of driving resistive loads as low as 300Ω or capacitances as large as 1000pF. The output common mode voltage is maintained at $VREF_{OUT}$ independent of the input common mode level. That is, the output amplifier PACell "re-references" the common mode level of the input signal. This is accomplished by continuously sensing the output common mode voltage and comparing it to $VREF_{OUT}$ as shown in Figure 3, and makes it possible to use an individual FilSum PACblock as a $VREF_{OUT}$ reference as discussed in the section titled "Using $VREF_{OUT}$".

Figure 3. Output $VREF_{OUT}$ Re-Referencing



Input Offset Auto-Calibration. A unique feature of the ispPAC20 is its ability to automatically calibrate itself to achieve very low offset error. This is done utilizing on-chip circuitry to perform an auto-calibration (auto-cal)

Theory of Operation (Continued)

sequence every time the device is turned on, or anytime it is commanded externally via the CAL pin or by a JTAG programming command. With this feature, the degradation of device offset performance that could occur over time and temperature is dramatically reduced. Specifically, this means one PACblock of an ispPAC20 in a gain configuration of one is guaranteed to never have an input offset error greater than 1mV, after being auto-calibrated. For higher gain settings when offset is especially important, the error is not multiplied by gain, but is instead divided by it, due to the unique architecture of the ispPAC20. When an individual PACblock is configured in a gain of ten, that results in an input referred offset error that never exceeds 100µV.

Internally, auto-calibration is accomplished by simultaneous successive approximation routines (SAR) to determine the amount of offset error referred to each of the two PACblock output amplifiers of the ispPAC20. That error is then nulled by a calibration DAC for each output amplifier. The calibration constant is not stored in E^2CMOS memory, but is recomputed each time the device is powered up or auto-cal is otherwise initiated. Initiation of auto-cal occurs when an ispPAC20 is powered on as part of its normal power on routine, or by a positive going pulse to the CAL pin, or by issuing the appropriate JTAG command.

During auto-cal, all ispPAC20 OA PACell outputs are driven to 0V and remain there until calibration is complete. The timing for the calibration process is generated internally. At power on, the sequence takes a maximum of 250ms, and when auto-cal is initiated via the CAL pin or by JTAG programming, it takes a maximum of 100ms to complete. The longer time required at power on insures the device power supply reaches its final value before calibration begins. Additional attempts to initiate auto-cal once calibration is in progress are ignored. Finally, the only direct indication of auto-cal completion will be the device's OA outputs returning to operational values from the 0V clamped state.

To insure maximum accuracy of the auto-cal procedure, all digital signals to the ispPAC20 should be suspended when calibration is in progress to avoid feed-through of noise to critical analog circuitry. This is especially true when auto-cal is initiated via JTAG command and the programming port is in use. There is sufficient time, however, to clock the JTAG controller back to its "reset" state without affecting the calibration process.

Bandwidth Trim. The bandwidth of an OA PACell is trimmed during manufacturing by adjusting the amplifier's feedback capacitance to optimize the step response. The trimmed step response resembles that of a critically damped system with minimum overshoot.

The bandwidth trim ensures a nominal feedback capacitance is always present, limiting the small signal bandwidth of an OA PACell to about 600kHz when configured in a gain of 1 (G=1). This should not be confused with the gain-bandwidth product of the op amp within the output amplifier PACells which is approximately 5MHz. It is important to note that the individual output amplifiers are always in essentially the same fixed gain configuration and do not, therefore, contribute to a decrease in signal bandwidth at higher PACblock gain settings. Since the gain of an individual PACblock is determined by varying the g_m of the input amplifier, bandwidth is not reduced in direct proportion to gain, as it would be in a traditional voltage feedback amplifier configuration. Specifically, small signal bandwidth is only reduced by a factor of 2, not the expected 10, with a PACblock gain setting change of G=1 to G=10. This is a significant advantage of the PACblock architecture.

Pole Accuracy Trim. Separate from the bandwidth trim capacitance, each FilSum PACblock contains a range of user selectable op amp feedback capacitance. This is made possible by a parallel arrangement of seven capacitors, each in series with an E^2CMOS switch. The user controls the position of the switches when selecting from the available capacitor values. The resulting capacitance is in parallel with the op amp feedback element, IAF, making 128 possible pole locations available. The capacitor values are not binarily weighted, instead they are chosen to optimize and concentrate pole spacing below 100kHz. There are 122 poles between 10kHz and 96kHz, which guarantees a step of no greater than 3.2% anywhere in that frequency range (to the nearest computed pole location). In fact, step size in over 50% of that range is less than 1.0%. Finally, capacitors are trimmed to achieve 5.0% accuracy (absolute) with regard to their nominal value.

PACblock Transfer Function

The block diagram for a PACblock is shown in Figure 1. The transfer function for a transconductor is:

$$I_P = - g_m \cdot V_{IN} \quad (1)$$

$$I_M = g_m \cdot V_{IN} \quad (2)$$

Using KCL (Kirchoff's current law) at the op amp inputs and assuming the input is connected to IA1 only:

Theory of Operation (Continued)

$$- V_{IN}\, g_{m1} + V_{OUT}\, g_{m3} + (V_{OUT+} - (V-))sC_F \quad (3a)$$

$$V_{IN}\, g_{m1} - V_{OUT}\, g_{m3} + (V_{OUT-} - (V+))sC_F \quad (3b)$$

where V- and V+ are the voltages at the op amp inverting and non-inverting inputs respectively. Because of feedback they are equal, so

$$\begin{aligned} -V_{IN}\, g_{m1} + V_{OUT}\, g_{m3} + (V_{OUT+} sC_F) \\ = V_{IN}\, g_{m1} - V_{OUT}\, g_{m3} + (V_{OUT-} sC_F) \end{aligned} \quad (4)$$

and the differential output voltage V_{OUT} is the difference $V_{OUT+} - V_{OUT-}$,

$$\frac{V_{OUT}}{V_{IN}} = \frac{g_{m1}}{g_{m3} + \frac{sC_F}{2}} \quad (5a)$$

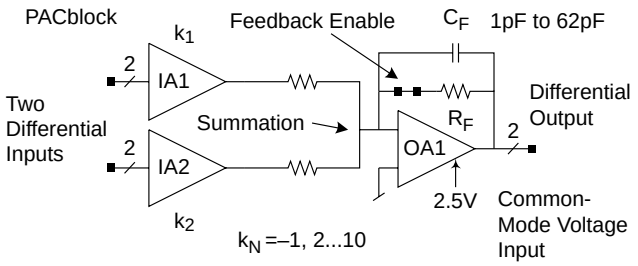
Since the PACblock has two separate inputs (IA1 and IA2) summed at the output amplifier input:

$$V_{OUT} = \frac{k_1 g_m V_{IN1} + k_2 g_m V_{IN2}}{g_{m3} + \frac{sC_F}{2}} \quad (5b)$$

The input amplifiers have a programmable gain of k·2µ/V (g_{m1} and g_{m2}) where k is an integer from -10 to 10. The feedback amplifier transconductance g_{m3} is fixed at 2µ/V, but may be disabled (g_{m3} = 0) to open-circuit the output amplifier's resistive feedback. The programmable feedback capacitance lies in the range 1pF to 62pF.

The PACblock model from PAC-Designer is shown in Figure 4. The output amplifier is configured as an inverting mode op amp and illustrates the summing configuration. The input instrument amplifiers are shown to make it clear that unlike a typical inverting op amp, the PACblock input impedance is extremely high. The input amplifier (IA) transconductance (gain) is shown as the value (k) above or below each amplifier. The gain of IA1 and IA2 are independently programmable. Because the feedback transconductor IAF (designated here as R_F) can be disabled by the user, a user configurable switch is shown in series.

Figure 4. PAC-Designer FilSum PACblock

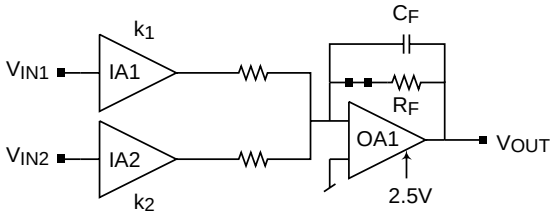


The FilSum PACblock implements two primary functions: the lossy integrator (low pass filter) and the integrator, both with gain.

Lossy Integrator. The lossy integrator's schematic within PAC-Designer is shown in Figure 5. Manipulating the PACblock transfer function of Equation 5 to better show the pole frequency yields:

$$V_{OUT} = \frac{k_1 V_{IN1} + k_2 V_{IN2}}{1 + \frac{sC_F}{2g_m}} \quad (6)$$

Figure 5. PAC-Designer PACblock Lossy Integrator



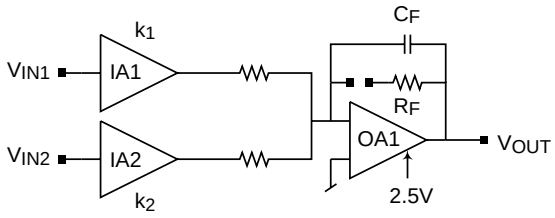
The DC gain of each input is set by k_1 or k_2 respectively, the gain constant for the input amplifiers. Below the pole frequency, this circuit can be viewed as a gain block. Because of the bandwidth trim capacitance, there is a minimum value of C_F causing the bandwidth to be approximately 550kHz when the DC gain is one. For larger gains, the input amplifier bandwidth begins to dominate the overall PACblock response, limiting the bandwidth to about 330kHz when the gain is 10.

Examining this transfer function shows the pole frequency is $(1/2\pi)(2g_m/C)$. Since g_m = 2µ/V and 1pF ≤ C_F ≤ 62pF, then 600kHz ≥ f_P ≥ 10kHz. Due to the selection options for feedback capacitance, there are at least 120 poles between 10kHz and 100kHz.

Theory of Operation (Continued)

Integrator. Switching out R_F (turning off IAF) removes the feedback element as shown in Figure 6. The integrator's transfer function can be derived from Equation 5b by setting $g_{m3} = 0$ (open circuit IAF (R_F)).

Figure 6. PAC-Designer PACblock Integrator (IAF Disabled; $g_{m3} = 0$)



$$V_{OUT} = \frac{k_1 V_{IN1} + k_2 V_{IN2}}{\frac{sC_F}{2g_m}} \quad (7)$$

The integrator slope is proportional to 1/f and, for the case of a single input, the transfer function magnitude equals |k| when the frequency is $(1/2\pi)(2g_m/C)$. The integrator should not be used as a stand-alone circuit element. It needs to be used in configurations that provide DC feedback to ensure the output does not saturate, as illustrated by the biquad filter circuit below.

Application Examples

Biquad Filter. By simply combining the two structures, the integrator providing feedback around the lossy integrator, creates a useful circuit. The block diagram is shown in Figure 7a and the schematic from PAC-Designer is shown in Figure 7b.

Figure 7a. Biquad Bandpass Filter Block Diagram

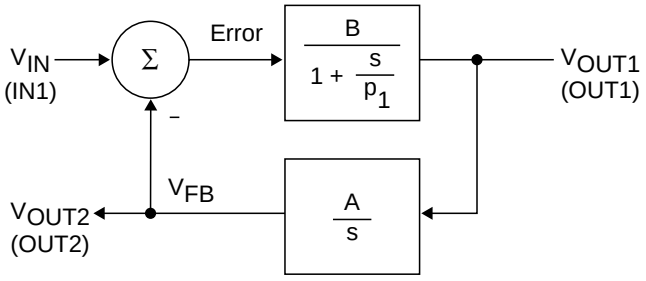
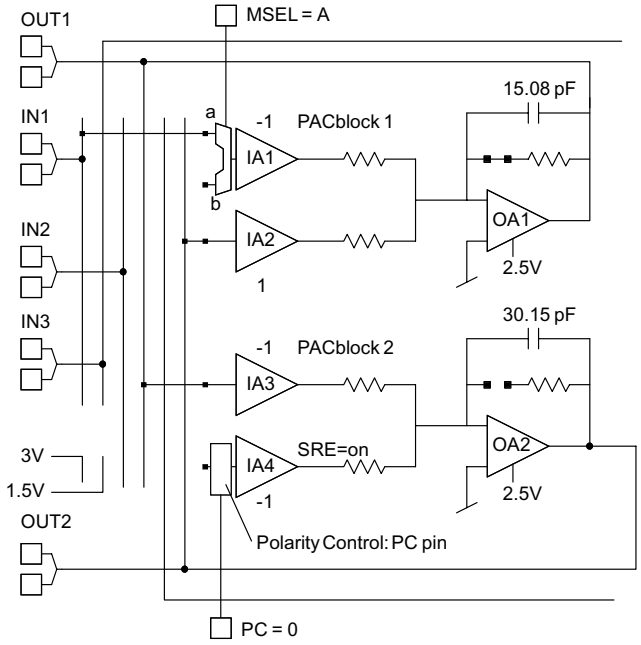


Figure 7b. Biquad Bandpass Filter Schematic

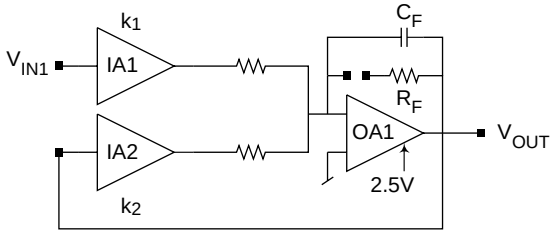


The transfer function OUT1(s)/IN1(s) is a band pass filter with programmable gain, Q and center frequency. Note the presence of DC feedback around the integrator. It can also be seen that the transfer function $V_{FB}(s)/V_{IN}(s)$ implements a lowpass filter. This application is discussed further in a separate application note.

To ease the design of Biquad Filters, PAC-Designer contains a macro tool that allows a user to simply specify filter corner, q factor and gain. This macro is accessed under the Tools menu.

Attenuator. The PACblock architecture makes variations possible on these two basic building blocks just described. An example uses summation to connect an input amplifier (IA2) in parallel with the feedback element (R_F), as shown in Figure 8.

Figure 8. PACblock $A_V < 1$



Theory of Operation (Continued)

The result is a circuit whose transfer function is:

$$\frac{V_{OUT}}{V_{IN}} = - \frac{k_1}{k_2 - \frac{sC_F}{2g_m}} \quad (8)$$

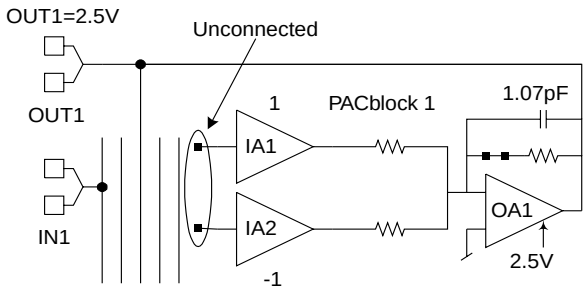
The gains k_1 and k_2 are independently set by the user. For stability, the phase of k_2 must be negative. The user can control the polarity of the transfer function by selecting the polarity of k_1. This circuit can either amplify or attenuate an input signal. The one in the denominator is due to R_F; if R_F is disabled, this term is eliminated. The level of attainable attenuation is as low as 1/11 (-20.8dB) with R_F enabled or 1/10 (-20dB) with R_F disabled.

When configuring a PACblock to attenuate, it is necessary to increase the value of feedback capacitance to maintain stability. Increasing feedback capacitance has the same beneficial effect as for a discrete op amp: It increases the network's phase margin which assists in maintaining stability.

Using $VREF_{OUT}$

The $VREF_{OUT}$ output is high impedance and it should be buffered when used as a reference. A PACblock can be made into a $VREF_{OUT}$ buffer as shown in Figure 9. The PACblock inputs are left unconnected and the feedback closed. In this condition the input amplifiers are tied to $VREF_{OUT}$ and the output amplifier's outputs are thus forced to $VREF_{OUT}$ or 2.5V. Either output is now a $VREF_{OUT}$ voltage source. This reference has the same drive capabilities of any ispPAC20 output. However, do not short the two outputs together. There is a small potential difference between them which will cause a steady state current to flow, thus needlessly dissipating power.

Figure 9. PACblock as $VREF_{OUT}$ Buffer

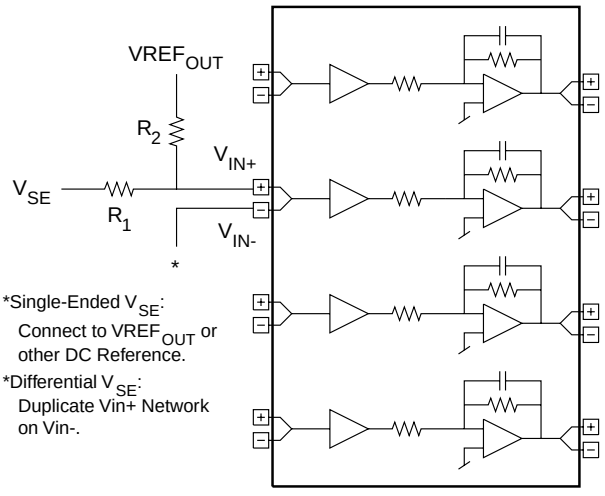


It is not always necessary to buffer the $VREF_{OUT}$ output. If it is used to reference a high impedance source, i.e., one that does not require more than 10µA, then it can be directly connected. An example is shifting the DC level of a signal connected to the input of a PACblock. In this case, the signal is AC coupled and "terminated" in $VREF_{OUT}$ through a minimum total resistance of 100kΩ. Referring to Figure 10b, if R_{IN} is greater than 200kΩ then the $VREF_{OUT}$ pin may be used without buffering.

Interfacing

When used in a single-supply system where the system common mode voltage is near $V_S/2$, signals may be directly connected to the ispPAC20 input. If the input signal does not have such a DC bias, then one needs to be added to the signal in order to accommodate the input requirements for the ispPAC20. A DC coupled bias can be added to a signal by using a voltage divider circuit as shown for one-half of the differential input in Figure 10a. Normally the choice for the reference DC voltage is the supply voltage, but other values may be used if necessary (and available).

Figure 10a. DC Biasing an Input Signal



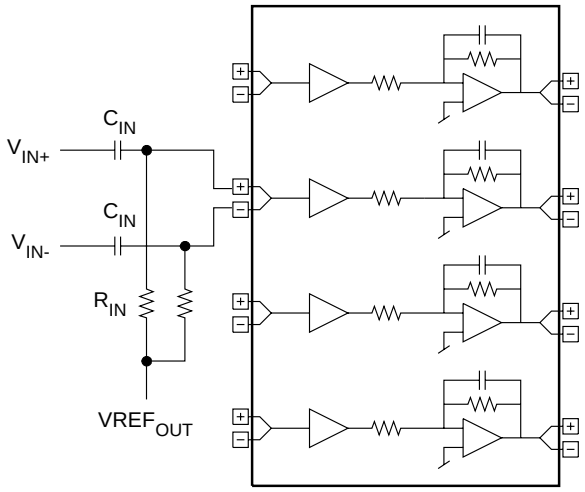
$$V_{IN+} = \frac{V_{SE}\ R_2}{R_1 + R_2} + \frac{VREF_{OUT}\ R_1}{R_1 + R_2}$$

Where DC coupling is not required, the input signal may be AC coupled as shown in Figure 10b. This circuit forms a high pass filter with a cutoff frequency of $1/(2\pi RC)$ and adds the necessary DC bias to the signal to accommodate the ispPAC20 input requirements. The DC reference should equal $V_S/2$, making $VREF_{OUT}$ the natural choice.

Theory of Operation (Continued)

The minimum resistance when using the $VREF_{OUT}$ buffer circuit of Figure 9 is 600Ω; when using the $VREF_{OUT}$ output pin it is 200kΩ (as discussed earlier).

Figure 10b. AC-coupled Input with DC Bias



Single-ended Operation

Single-ended signals may be connected to the ispPAC20 input and one of the two differential ispPAC20 outputs can be used to drive single-ended circuitry. So, in addition to fully differential I/O, either the input, output or both may be used single-ended.

Single-ended Input. To connect the ispPAC20 differential input to a single-ended signal, one of the differential inputs needs to be connected to a DC bias, preferably $VREF_{OUT}$. The input signal must either be AC coupled (as in Figure 10b) or have a DC bias equal to the DC level of the other input. Since the input voltage is defined as V_{IN+}- V_{IN-}, the common mode level is ignored. The signal information is only present on one input, the other being connected to a voltage reference.

Single-ended Output. Connecting the output to a single-ended circuit is simpler still. Simply connect one-half of the differential output, but not the other. Either output conveys the signal information, just at half the magnitude of the differential output. The DC level of the single-ended output will be $VREF_{OUT}$ due to the re-referencing aspect of the FilSum PACblock. If the load is not AC coupled and is at a DC potential other than $VREF_{OUT}$, the load draws a constant current. Using one of the differential outputs halves the available output voltage swing ($3V_{PP}$ versus $6V_{PP}$) and since the output current capacity is the same whether driving differentially or single-ended, a single output can drive twice the load as the differential output (150Ω vs. 300Ω or 2000pF vs. 1000pF). If the load requires DC current, the amount available for voltage swing is reduced. The output is capable of 10mA, so any DC current raises the minimum allowable load impedance.

Noise vs. Gain

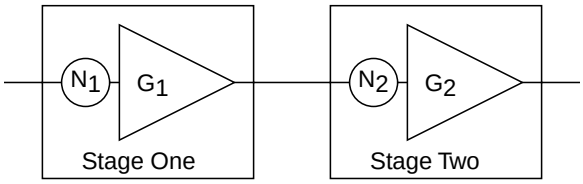
Noise gain is the gain of a circuit configuration to its combined input-referred circuit noise. The noise gain of an inverting op amp circuit is:

$$\text{Noise Gain} = 1 + \text{Closed Loop Voltage Gain} \qquad (9)$$

In this case, the noise gain of the circuit increases proportionally to the circuit gain.

A FilSum PACblock contains an input amplifier stage followed by an output amplifier. In this way it can be viewed as a system, with each of the components having its own contribution to the overall noise as shown in Figure 11. Both the output amplifier noise (N_2) and input amplifier noise (N_1) contribute to the overall noise performance, but the contribution due to the output amplifier dominates except at input gains near 10. The result is that the SNR of a FilSum PACblock is nearly constant versus gain. This is different than the behavior predicted by Equation 9.

Figure 11. Multistage ispPAC Noise Diagram



G_2 = Constant

$$\text{Output Noise Voltage} = G_1 G_2 \sqrt{N_1^2 + \left(\frac{N_2}{G_1}\right)^2} \qquad (10a)$$

If $N_2/G_1 > 3 \cdot N_1$, then

$$\text{Output Noise Voltage} \cong G_2 N_2 \qquad (10b)$$

There is a few dB decrease in SNR as the gain approaches 10. This characteristic implies the input amplifier noise contribution is approaching that of the op amp. As the gain of the input amplifier nears 10, its noise contribution in Equation 10a (N_1) approaches that of the op amp

Theory of Operation (Continued)

and becomes a factor in the overall output noise voltage, causing it to increase.

Input Common-Mode Voltage Range

For the ispPAC20, both maximum input signal range and corresponding common-mode voltage range are a function of the input gain setting. The maximum input voltage times the gain of an individual PACblock cannot exceed the output range of that block or clipping will occur. The maximum guaranteed input range is 1V to 4V, with an extended typical range of 0.7V to 4.3V for a 5V supply voltage.

The input common-mode voltage is $V_{CM} = (V_{CM+} + V_{CM-})/2$. When the value of V_{CM} is 2.5V, there are no further input restrictions other than the previously mentioned clipping consideration. This is easily achieved when the input signal is true differential and referenced to 2.5V.

When V_{CM} is not 2.5V and the gain setting is greater than one, distortion will occur when the maximum input limit is reached for a particular gain. The lowest V_{CM} for a given gain setting is expressed by the formula, $V_{CM-} = 0.675V + 0.584G \cdot V_{IN}$ where G is the gain setting and V_{IN} is the peak input voltage, expressed as $|V_{IN+} - V_{IN-}|$ and the highest V_{CM} is $V_{CM+} = 5.0V - V_{CM-}$ where 5V is the nominal supply voltage.

In Table 1, the maximum V_{IN} for a given V_{CM-} to V_{CM+} range is given. If the maximum V_{IN} is known, find the equivalent or greater value under the appropriate gain column and the widest range for V_{CM} will be found horizontally across in the left-most two columns. Only a V_{CM} range equal to or less than this will give distortion-free performance. Conversely, if the maximum V_{CM} range is known, the largest acceptable peak value of V_{IN} can be found in the corresponding gain column. All values of V_{IN} less than this will give full rated performance.

Table 1. Input Common-Mode Voltage Range Limitations

Input Voltage Magnitude (Volts-Peak)											
V_{CM-}	V_{CM+}	G=1	G=2	G=3	G=4	G=5	G=6	G=7	G=8	G=9	G=10
1.000	**4.000**	0.557	0.278	0.186	0.139	0.111	0.093	0.080	0.070	0.062	0.056
1.100	**3.900**	0.728	0.364	0.243	0.182	0.146	0.121	0.104	0.091	0.081	0.073
1.200	**3.800**	0.899	0.450	0.300	0.225	0.180	0.150	0.128	0.112	0.100	0.090
1.300	**3.700**	1.071	0.535	0.357	0.268	0.214	0.178	0.153	0.134	0.119	0.107
1.400	**3.600**	1.242	0.621	0.414	0.310	0.248	0.207	0.177	0.155	0.138	0.124
1.500	**3.500**	1.413	0.707	0.471	0.353	0.283	0.236	0.202	0.177	0.157	0.141
1.600	**3.400**	1.584	0.792	0.528	0.396	0.317	0.264	0.226	0.198	0.176	0.158
1.700	**3.300**	1.756	0.878	0.585	0.439	0.351	0.293	0.251	0.219	0.195	0.176
1.800	**3.200**	1.927	0.964	0.642	0.482	0.385	0.321	0.275	0.241	0.214	0.193
1.900	**3.100**	2.098	1.049	0.699	0.525	0.420	0.350	0.300	0.262	0.233	0.210
2.000	**3.000**	2.270	1.135	0.757	0.567	0.454	0.378	0.324	0.284	0.252	0.227
2.100	**2.900**	2.441	1.220	0.814	0.610	0.488	0.407	0.349	0.305	0.271	0.244
2.200	**2.800**	2.612	1.306	0.871	0.653	0.522	0.435	0.373	0.327	0.290	0.261
2.300	**2.700**	2.783	1.392	0.928	0.696	0.557	0.464	0.398	0.348	0.309	0.278
2.400	**2.600**	2.955	1.477	0.985	0.739	0.591	0.492	0.422	0.369	0.328	0.295
2.426	**2.574**	3.000*	1.500*	1.000*	0.750*	0.600*	0.500*	0.429*	0.375*	0.333*	0.300*
2.500	**2.500**	3.126	1.563	1.042	0.782	0.625	0.521	0.447	0.391	0.347	0.313

*Peak input voltage for guaranteed performance at a given gain setting.

Theory of Operation (Continued)

DAC PACell

The ispPAC20 contains an 8-bit, voltage output, digital-to-analog converter (DAC) PACell with many unique features and options. Interface modes are user selectable and include a direct 8-bit parallel port, a serial JTAG address mode, or serial SPI address mode. The output of the DAC is fully differential, making it compatible with the rest of the ispPAC20's internal analog I/O. The DAC's voltage output is available via external pins as well as by on-chip routing for optional internal connection to either the comparator PACells or any of the instrument amplifier input PACells.

DAC Data Input Coding

Data input to the DAC, whether in serial or parallel mode, determines its output value. The coding of the DAC is in straight binary and corresponds to input to output relationship shown in Table 2, DAC I/O. In all serial modes, 8 bits of data are clocked in with D0 (the LSB) being first in the data stream and D7 (the MSB) being last.

DAC Address Modes

Addressing modes are controlled from within PAC-Designer (options in the DAC port configuration pop-up) and by two external pins (DMode and ENSPI). Figure 12 diagrams the various input data paths used to implement the various ispPAC20 DAC addressing modes. Also included in the figure is a truth table of the user E^2 settings and input logic levels required to enable them. All serial data input modes are 8 bits long and clocked in LSB (D0) first.

The choice of addressing modes depends largely on application needs, but the primary benefit of each addressing mode is as follows:

JTAG/E^2: Power-up state of DAC is determined by E^2 configuration memory. The DAC input code can still be changed, but only by reprogramming the E^2 memory via JTAG command and subject to the maximum number of programming cycles allowed. This is the preferred mode to use when the DAC setting must be retained when device power has been cycled off and then on again.

Parallel: This mode allows direct parallel update access to the DAC. The DAC can be updated continuously without affecting E^2 programming cycle endurance issues. The DAC E^2 configuration cells can still be programmed via serial JTAG commands directly from the value stored in the parallel input data latches at any time, if desired.

JTAG/Direct: The DAC can be addressed directly, bypassing the E^2 configuration memory via the standard JTAG serial interface protocol. Using this serial addressing mode retains the ability to reprogram the ispPAC20 DAC at any time without having to reconfigure the interface from one mode to another.

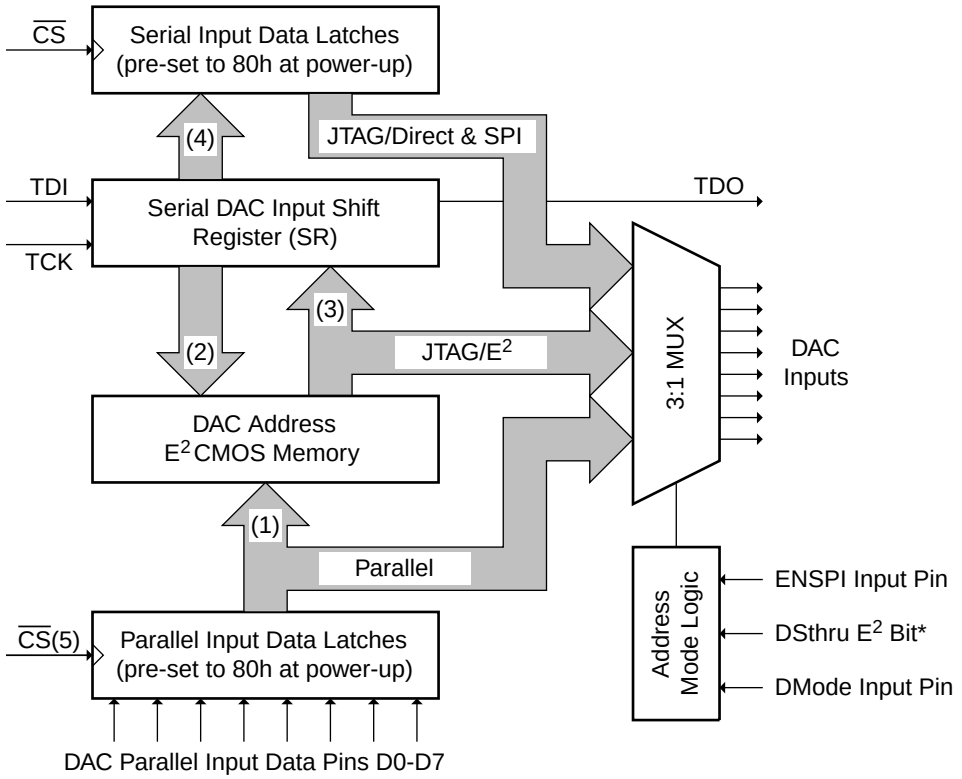
SPI: The DAC can be addressed directly, bypassing the E^2 configuration memory via an SPI compatible serial interface protocol. The SPI serial interface is one of the most widely used protocols for communication with mixed signal devices of all types. While in the SPI addressing mode, programming of the DAC E^2 configuration memory is not possible.

Table 2. DAC I/O

	Code		Nominal Voltage		
	DEC	**HEX**	**Vout+ (V)**	**Vout- (V)**	**Vout (Vdiff)**
-Full Scale (-FS)	0	00	1.0000	4.0000	-3.0000
	32	20	1.3750	3.6250	-2.2500
	64	40	1.7500	3.2500	-1.5000
	96	60	2.1250	2.8750	-0.7500
MS - 1LSB	127	7F	2.4883	2.5117	-0.0234
Mid Scale (MS)	128	80	2.5000	2.5000	0.0000
MS + 1LSB	129	81	2.5117	2.4883	0.0234
	160	A0	2.8750	2.1250	0.7500
	192	C0	3.2500	1.7500	1.5000
	224	E0	3.6250	1.3750	2.2500
+Full Scale (+FS)	255	FF	3.9883	1.0117	2.9766
LSB Step Size			x + 0.0117	x - 0.0117	0.0234
+FS + 1LSB			4.0000	1.0000	2.9766

Theory of Operation (Continued)

Figure 12. ispPAC20 DAC Interface Options



Registers updated after JTAG command(s):
(1) DBE, PrgDAC (DMode=1)
(2) AddDAC, DBE, PrgDAC (DMode=0)
(3) VerDAC
(4) AddDAC (E^2 bit DSthru=1, ENSPI=0)
SPI mode only: Rising edge of CS
(5) Rising edge of CS (only if DMode=1)

Address Mode	ENSPI Pin	DSthru E^2 Bit	DMode Pin
JTAG/E^2	0	0	0
Parallel	0	0	1
JTAG/Direct	0	1	X
SPI	1	X	X

*Decoded from the user selection in the DAC port configuration pop-up.

Table 3. DAC Address Modes

Action	JTAG/E^2 Serial Mode	Parallel Mode	JTAG/Direct Serial Mode	SPI Serial Mode
E^2 Cells Programmed Via:	Serial Input SR	Parallel Latches	Serial Input SR	No E^2 Access
DAC Input Comes From:	E^2CMOS Memory	Parallel Latches	Serial Latch	Serial Latch
DAC Updated On:	During Update-DR, falling edge TCK (1)	Rising Edge CS	During Update-DR, falling edge TCK	Rising Edge CS
Rising Edge CS Updates:	Serial Latch	Serial Latch Parallel Latch	Serial Latch	Serial Latch
DAC Output at Power-Up:	Stored E^2 Value	80h (Vout+, Vout-=2.5V)	80h (Vout+, Vout-=2.5V)	80h (Vout+, Vout-=2.5V)
TDO Serial Output in Hi-Z State During JTAG AddDAC Operation	If E^2 bit DisTDO =1, Otherwise active during Shift-DR/IR JTAG state	No TDO if TCK pin is not clocked	If E^2 bit DisTDO =1, Otherwise active during Shift-DR/IR JTAG state	When CS is high

Notes: (1) DAC output goes from –FS to +FS during E^2 programming cycle (JTAG DBE or DAC Bulk Erase, and PrgDAC or Program DAC commands) before settling to the final input code value.

Theory of Operation (Continued)

DAC Address Mode Details

DAC Parallel Mode Addressing. The parallel addressing mode uses the eight external (D0-D7) data pins of the ispPAC20 to address the DAC. The DMode (DAC E^2/parallel Mode) logic input pin determines whether the input data path is routed from E^2 memory (DMode =0) or directly from the parallel input data pins (DMode =1). In addition, both serial input modes (JTAG/Direct and SPI) must be disabled to access the parallel input mode. This means the shift register option in the DAC port configuration pop-up is selected, the ENSPI (Enable SPI serial mode) logic input pin is low and the DMode logic input pin is high. Data is latched into the parallel data latches on a positive going edge of CS (Chip Select) and the output of the DAC changes to its new value at this time according to the setup timing constraints in the AC specification waveform tables. When a device is first turned on, the parallel data latches are initialized to code 80h, which corresponds to 2.5V on both DAC analog output pins. To otherwise start up with the value DAC code programmed in E^2 memory (instead of the default 80h), the DMode logic input pin must remain low until the first data update of the parallel input data latch at which time the contents of the DAC reflect the parallel data input pins.

JTAG/E^2 Serial Mode Addressing. The JTAG/E^2 serial mode is the only addressing mode where the ispPAC20 powers up with the DAC set to the input code stored in its internal E^2 configuration memory. In all other modes the DAC defaults to input code 80h (2.5V on both output pins) at turn on. The DAC can be changed while in this mode, but only by a process of reprogramming the DAC E^2 memory cells themselves via routine JTAG commands. This is sometimes desirable when a particular DAC output operating point is reached that the system is then required to “remember”. This update can be accomplished via programming the DAC directly through the JTAG interface of the ispPAC20 without perturbing any of the rest of the chip’s function or operation.

It should be noted, however, that the DAC outputs are directly determined by the state of their E^2 configuration memory. That means if the DAC E^2 cells are reprogrammed to change codes, the DAC output will follow the E^2 transition states until their final programmed value is reached. A DAC E^2 programming cycle consists of an erase during which the output goes to minus full-scale (-FS), then a write during which the output briefly goes to plus full-scale (+FS) before the E^2 cells transition to their final programmed values and the output settles there as well. This phenomenon only applies when in the JTAG/E^2 serial address mode. In all other addressing modes, the DAC changes to its new value immediately after a latch register is clocked.

JTAG/Direct Serial Mode Addressing. Unlike the previous method of addressing the ispPAC20 DAC from the E^2 cells directly, JTAG/Direct serial mode interfaces the DAC via the serial input data latches. After a data word is shifted into the serial input shift register via JTAG command (AddDAC), the DAC is immediately updated on the falling edge of clock TCK in the UpdateDR state. The E^2 cells are bypassed entirely in this mode. The advantages are that the DAC can be addressed separately from the rest of the ispPAC20 via the serial JTAG interface and can be continuously updated an unlimited number of times. The serial data rate of 5MHz is much faster than the settling time of the DAC making this an acceptable way of addressing and changing the output for full speed AC applications. It is, of course, also suitable for applications where the DAC output needs to be varied from time to time, and the need to store the last code before power down on-chip is not critical.

SPI Serial Mode Addressing. Finally, the ispPAC20 can be addressed using a serial interface mode that is compatible with the industry standard SPI protocol (serial peripheral interface, a Motorola trademark). Like the JTAG/Direct serial mode, the DAC E^2 configuration is bypassed in SPI serial mode allowing the DAC to be updated continuously and for an unlimited number of cycles if desired. Whenever the ENSPI (enable SPI) pin is high, the ispPAC20 is in the SPI serial addressing mode and the 8 bits of DAC input data can be clocked in with D0 (the LSB) being first in the data stream and D7 (the MSB) being last, if the device is selected by the CS (chip select) pin being low. The data is latched in and the DAC output changes on a subsequent rising edge of CS.

Comparator PACell Operation

The ispPAC20 has two programmable, double difference comparator PACells on chip that include many user programmable options to optimize their utility. These comparators operate no differently than any standard comparator, that is whenever the +(plus) input is positive with respect to the -(minus) input, its logic output will be high, otherwise they will be low. Unlike most other available comparators, however, inputs to the ispPAC20 comparator PACells are fully differential (true double-difference comparators). Both the plus and minus inputs of the ispPAC20 comparators have a Vin+ and a Vin- with the differential input voltage defined as [(Vin+) - (Vin-)]. This means the comparator output is high whenever the

Theory of Operation (Continued)

differential voltage on the +(plus) input is positive with respect to the differential input voltage on the -(minus) input.

Comparator Input Options

All inputs to the comparators can be accessed from several different points including signals external to the ispPAC20. When first shown in the PAC-Designer software design entry screen, the inputs to the comparators appear not to be connected to any signal source. In fact, whenever no connection is indicated, the Vin+ and Vin- lines (denoted by a single line in PAC-Designer) are both connected to 2.5V DC. That means that if the minus input were left unconnected in PAC-Designer, the differential voltage on that input would be 0V (2.5V - 2.5V = 0V). At this point any positive differential voltage on the plus input of that comparator would result in a logic 1 output, and any negative a logic 0.

The output of PACblock 2 (OA2) is available to any input of CP1 or CP2 as is the external input pin, IN3. In the case of a signal on IN3, it could be routed to one of the PACblocks as well as the comparators to control a switching threshold or other level determined event. Using IN3 as a standalone input going only to one of the comparators and the CPIN pin (comparator external input), both the plus and minus inputs to the comparators could come from entirely external signals.

The most common source for deriving reference levels on the comparators would be directly from the internal 8-bit DAC. In addition to the 256 voltage levels being directly available from the DAC, a constant 1.5V and 3.0V is also available for setting a comparator input threshold. These fixed values free the DAC to be used for other circuit purposes such as nulling system offset voltages or programming ADC reference inputs.

It should be noted that the plus input path of CP2 effectively performs a negation of the differential voltage to that input (denoted by an additional inversion symbol in PAC-Designer). The utility of this operation is that an identical differential signal can be applied to the plus inputs of both comparator PACells and result in a symmetrical window about 2.5V. For example if the +1.5VDC input line is connected to both comparator plus inputs, CP1's plus input is +1.5V differential, and CP2's plus input is then -1.5V differential. If both minus inputs were both connected to CPin in PAC-Designer (the external comparator input pin), the result would be a logic 1 on CP1 when the external input was below +1.5Vdiff and a logic 1 on CP2 whenever it was above -1.5Vdiff. Furthermore, the WINDOW (window compare output pin) which is the exclusive OR of the two CPout pins would result in a logic 0 any time the signal was between +-1.5Vdiff on the external input and a logic 1 anytime it was outside that window.

Optional Comparator Hysteresis

Another programming option provided for the user is the ability to enable or disable comparator hysteresis. Hysteresis is useful in situations where a slow moving signal, or an uncertain transition condition exists that would otherwise result in excessive noise on the comparator output. The magnitude of this hysteresis is nominally 47mV and can be either enabled or disabled in E^2 configuration memory and concurrently affects both comparators. It is symmetrical with respect to any input change, which means that regardless of which direction the input causing the state change comes from (with respect to the reference input), it will have to change at least 47mV above or below the reference to cause another output state change. The default initial condition of the hysteresis setting is on. Comparator hysteresis can be disabled by selecting the appropriate edit symbol command in PAC-Designer and making the change.

Polarity Control of IA4

Normally the gain and polarity for an individual IA (instrument amplifier input) PACell is chosen from the range of choices from -10 to +10 (in integer steps) directly in PAC-Designer from a single gain setting listbox choice. With the ispPAC20 this is the case for IA1, IA2 and IA3. IA4 on the other hand, only has gain choices from -10 to -1 available in this particular dialog box. The reason is the positive gains are actually realized by internally reversing the polarity of the differential inputs, effectively multiplying the ten negative gains by -1 to achieve the positive gain values. With IA4, the control of this inversion "routing" switch has been made externally available for some unique device operating modes. The control is made through the external PC (polarity control) pin, or signals routed internally to this same input pin. See Table 4 and Figure 13 for complete PC pin operation details.

These comparator logic control/option modes are all configured within PAC-Designer to achieve the operation summarized in Table 4 and Figure 13. More information is available in the online help file for PAC-Designer and in application notes that describe the circuits made possible by this on-chip logic.

Theory of Operation (Continued)

Table 4. Comparator Logic Control Modes/Options

Mode	PC Pin Function	Description
Fixed, Non-Inverting	None (internal).	Always generates a +1 times whatever the gain setting of IA4 is. IA4 can be set to gains of -1 to -10 in this mode. Display of the gain setting for IA4 in PAC-Designer is of the correct polarity.
PC Direct	Direct control of IA4 polarity via the PC pin, logic 0 = inverted, logic 1 = no inversion (external).	IA4 gain setting is correct as shown in PAC-Designer (-1 to -10) if PC pin input equals a logic 1 (no inversion). If the PC pin input equals a logic 0, the setting of IA4 will be inverted with respect to what is displayed in PAC-Designer. Terminating the PC pin low (externally) in this mode is the most direct way of achieving a constant setting of positive gain (+1 to +10) for IA4.
RS Flip-Flop	Clamps OA2 to VREFout when PC is a logic 1, and has no effect when it's a logic 0 (external).	Both comparators combine to generate a set/reset function on the WINDOW logic output pin instead of the usual XOR function. This signal is also routed internally to IA4 for polarity control. When PC is a logic 0, CP1 positive transitions generate a set command and positive transitions of CP2 a reset command. For example, if PACblock 2 is configured as an integrator and its output is fed to CP1 and CP2 (configured for window comparison), a voltage controlled oscillator will result from the RS Flip-Flop set/reset reversing the polarity every time the integration exceeds the upper and then lower window boundaries in sequence. When PC is a logic 1, the output of OA2 goes immediately to 2.5V and stays in "hold" mode until PC returns to a logic 0. This effectively implements a gated oscillator function.
CP1 Direct	None. IA4 polarity control based directly on CP1 output, a logic 0 = inverted, logic 1 = no inversion (internal).	In this mode, the output of comparator 1 (CP1) controls the polarity of IA4, a logic 1 = no inversion, a logic 0 = inverted. When the output of CP1 is in the direct mode, the polarity control is all internal. If the CP1 Buffer E2 bit is set, CP1 changes can only occur if clocked by the PC pin externally (rising edge).
PC Clock	Clocks CP1 data register (external).	Always enabled anytime the CP1Buffer E2 configuration bit is set. Each rising edge of PC clocks whatever data is read from CP1's output into its output register regardless of what other function is being performed by the PC pin. In certain modes listed above, the operation of CP1 in buffered mode combined with the need to clock its output using the PC pin, would interfere with or prevent the proper operation of some circuit function implementations.

IA4 Slew-Rate Enhancement

Because of the special applications addressed by the inclusion of a polarity control function in IA4, its circuitry has been modified to include a slew-rate enhancement feature. This circuitry is not part of the output amplifier (OA) PACell and therefore does not change the specified slew rate as given in the data sheet. Rather, it enhances the operation of IA4 and improves its performance in applications such as a voltage controlled oscillators (VCO), thereby improving its performance in reproducing non-linear transfer functions. The ispPAC is shipped with this bit normally enabled. If identical operation between all IA PACells is desired, the SRE bit associated with IA4 can be disabled by selecting the appropriated edit symbol command in PAC-Designer and making the change.

Multiplexer Input control of IA1

An external multiplexer select (MSEL) pin is provided that controls which of two possible input connections are routed to the input of IA1. When MSEL is a logic 0, input line "A" is selected to go to IA1, when it is a logic 1, input line "B" is used. This arrangement allows a number of applications to be implemented, from something as straight forward as two input signal channels, to more complex functions such as those provided for by the polarity control pin (with the option of different signals being used as well).

Theory of Operation (Continued)

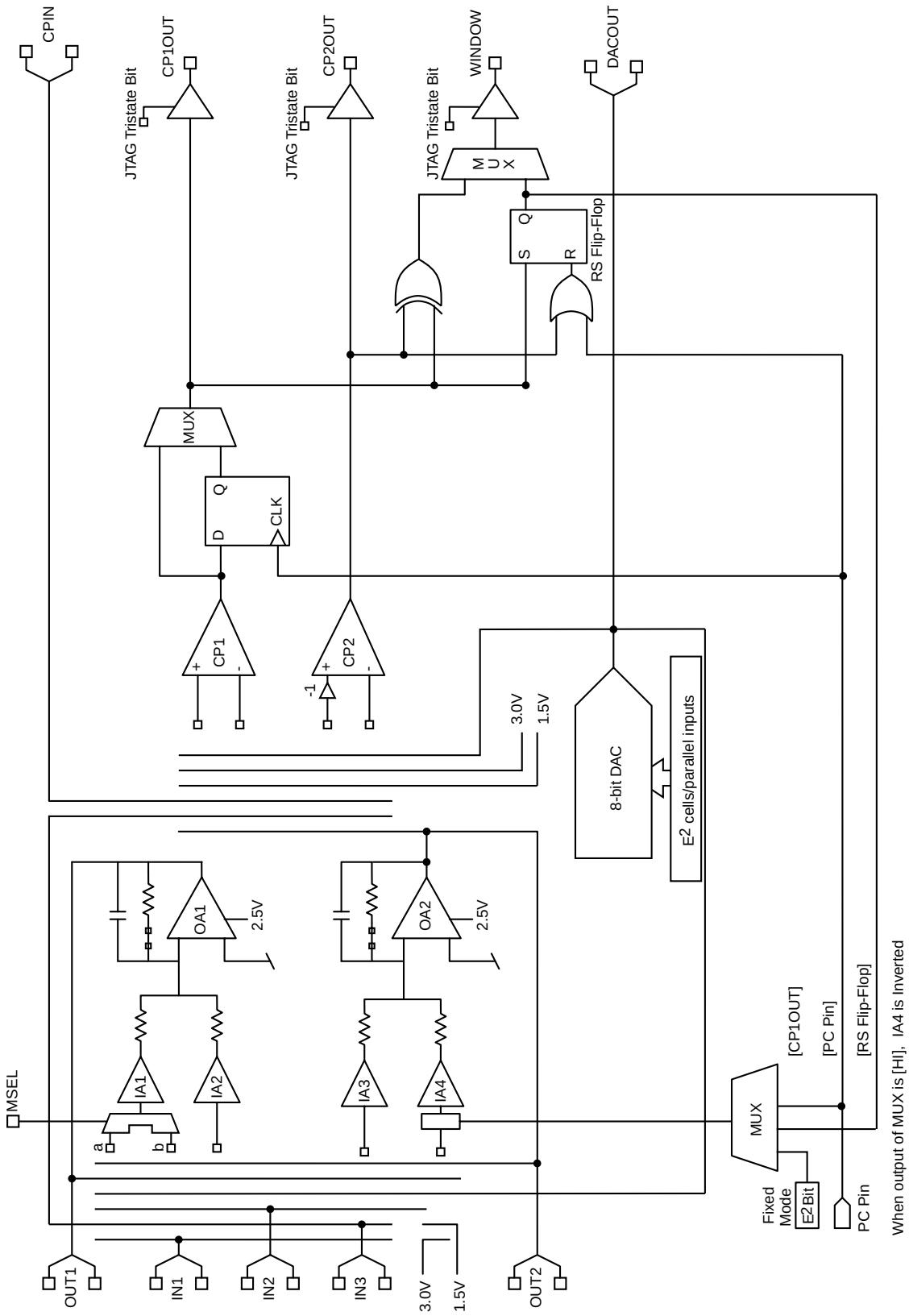
Table 5. JTAG User Configuration Bits

Symbol	Name	Description
DSthru	Direct Serial Pass-Thru	Used to enable the serial, JTAG/Direct mode. Enables addressing of DAC directly from the serial latches instead of the E^2 cells. Overrides the effect of the DMode pin. Overridden itself by the ENSPI mode pin (high).
CPHyst	Comparator Hysteresis	Used to enable comparator hysteresis mode for both comparators (47mV). Selected in PAC-Designer by double clicking on the hysteresis symbol in between the two comparators or by using the edit symbol dialog.
DisTDO	Disable TDO	Used to disable the TDO output, or in other words place it in permanent high-impedance output mode. This is done to reduce unnecessary on-chip perturbation of the analog circuitry while changing the DAC codes in either the JTAG/E^2 or JTAG/Direct modes. Has no effect on TDO when ENSPI is high and the DAC is in SPI mode. Note that TDO is disabled at certain times even when the DisTDO bit is not set.
PCMode1, 2	Polarity Control Mode	Used to control the various modes of PC (polarity control) digital input pin function. These include simple logic control of IA4's gain polarity, a blocking of the PC pin input altogether, an oscillator flip-flop control mode and gating of the oscillator flip-flop mode, and direct connection to CP1OUT.
WCMode	Window Compare Mode	Used to enable either XOR or FF output mode on the Window output pin.
CP1Buffer	Comparator 1 Latch Enable	Causes output of Comparator 1 (CP1) to be latched into a D-flip-flop before being output. Latch is updated by clocking the PC input pin.
UES1-7	User Electronic Signature	These bits are available to store information about an individual device in on-chip E^2 configuration memory. For example, the configuration code, performance data or other classification data could be stored and later retrieved to identify some unique property associated by the user with the device.
CPOut	Comparator Output Disable	Disables all three comparator-related outputs (CP1, CP2 and Window) placing them in a high impedance state. The purpose of the option is to allow quieter operation of comparators (less effect on other analog circuitry) when their outputs are only required for on-chip operation.
SRE	Slew Rate Enhancement	Normally on, this bit enhances the slew rate capability of IA4. Normally this is of greatest benefit in such applications as voltage controlled oscillators where an improvement in non-sinusoidal waveform generation is desired. Has no effect on THD of normal signals, but can still be disabled if output needs to be matched exactly to the characteristics of IA1-3.
ESF	Electronic Security Fuse	Setting this bit causes all subsequent readouts of the device configuration to be disabled (JTAG Verify commands). Can be reset by performing a JTAG user bulk erase command and reprogramming the device. This feature is used to prevent unauthorized readout of the device's configuration.
EnCMVin1, 2	Enable External CMV Ref	Enables an external input reference to determine the output common-mode voltage of OA1 and/or OA2 instead of the normally-used 2.5V from on-chip.

JTAG User Bits

There are a number of user-configured E^2 bits that control various aspects of ispPAC operation and can all be accessed in either the pull-down menus or directly in the schematic design entry screen of the PAC-Designer software. See the online help associated with the ispPAC20 in PAC-Designer for more details of how to set/program various operation modes. The list of control E^2 bits available is given in Table 5.

Figure 13. PAC-Designer Design Entry Screen With Detailed Logic Schematic Diagram



Software-Based Design Environment

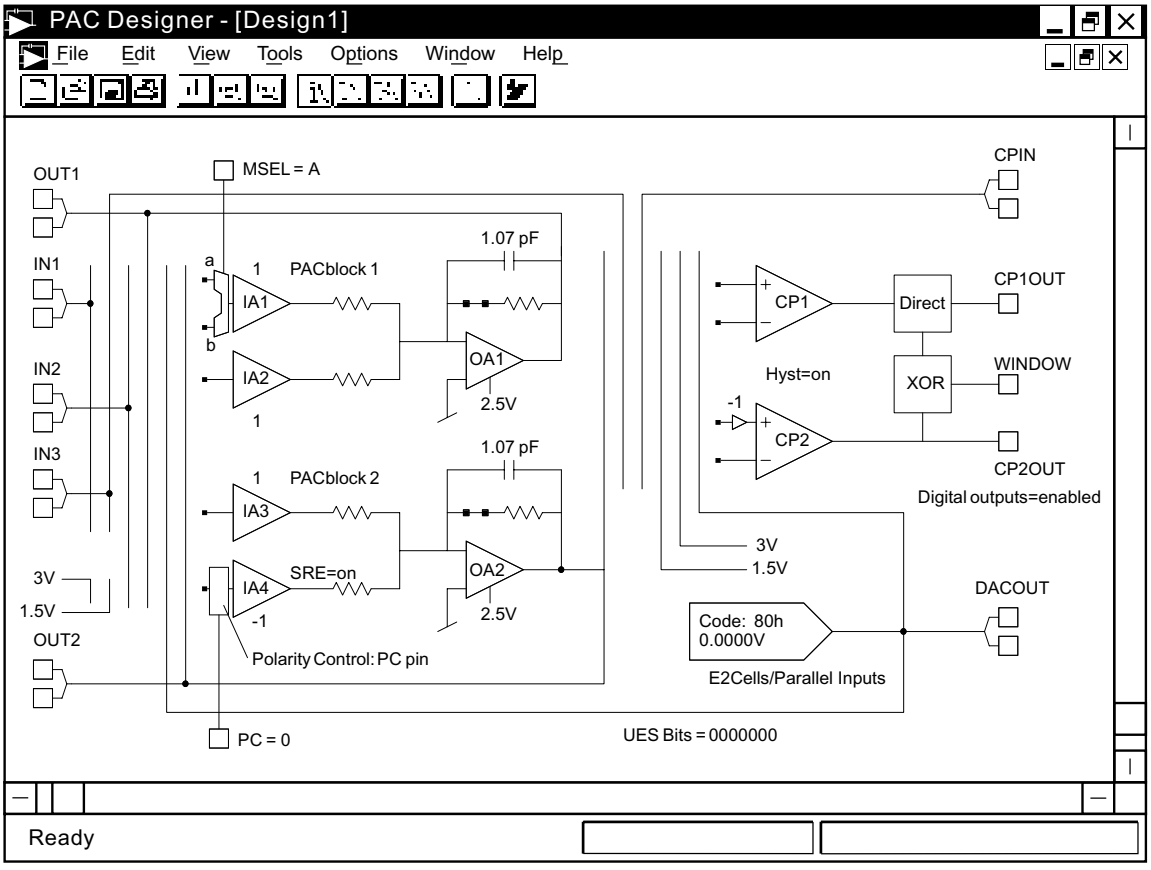
Design Entry Software

Designers configure the ispPAC20 and verify its performance using PAC-Designer, an easy to use, Microsoft Windows compatible program. Circuit designs are entered graphically and then verified, all within the PAC-Designer environment. Full device programming is supported using PC parallel port I/O operations and a download cable connected to the serial programming interface of the ispPAC20. A library of configurations is included with basic solutions and examples of advanced circuit techniques. In addition, comprehensive on-line and printed documentation is provided that covers all aspects of PAC-Designer operation.

The PAC-Designer schematic window, shown in Figure 14, provides access to all configurable ispPAC20 elements via its graphical user interface. All analog input and output pins are represented. Static or non-configurable pins such as power, ground, $VREF_{OUT}$, and the serial digital interface are omitted for clarity. Any element in the schematic window can be accessed via mouse operations as well as menu commands. When completed, configurations can be saved, simulated, and downloaded to devices.

PAC-Designer operation can be automated and extended by using custom-designed Visual Basic® programs that set the interconnections and the parameters of ispPAC products. These stand-alone programs are called macros. An example of such a macro is the biquad filter generator supplied with PAC-Designer. With this macro, filter parameters such as gain, Q and corner frequency are input directly and then automatically converted to a schematic configuration. More information on this and other topics is included in the on-line documentation as well as the *PAC-Designer Getting Started Manual*.

Figure 14. Initial PAC-Designer Schematic Design Entry Screen



In-System Programmability

Electronic Security

An electronic security "fuse" (ESF) bit is provided in every ispPAC20 device to prevent unauthorized readout of the E^2CMOS user bit patterns. Once programmed, this cell prevents further access to the functional user bits in the device. This cell can only be erased by reprogramming the device, so the original configuration can not be examined once programmed. Usage of this feature is optional.

Production Programming Support

Once a final configuration is determined, an ASCII format JEDEC file is created using the PAC-Designer software. Parts can then be ordered through the usual supply channels with the user's specific configuration already preloaded into the parts. PAC-Designer will also export an SVF file which can be used with ispVM™ for embedded programming applications. By virtue of its standard interface, compatibility is maintained with existing production programming equipment giving customers a wide degree of freedom and flexibility in production planning. Other options exist for production programming, including a C-coded library of ispVM functions. Contact ispPACs@latticesemi.com for more details.

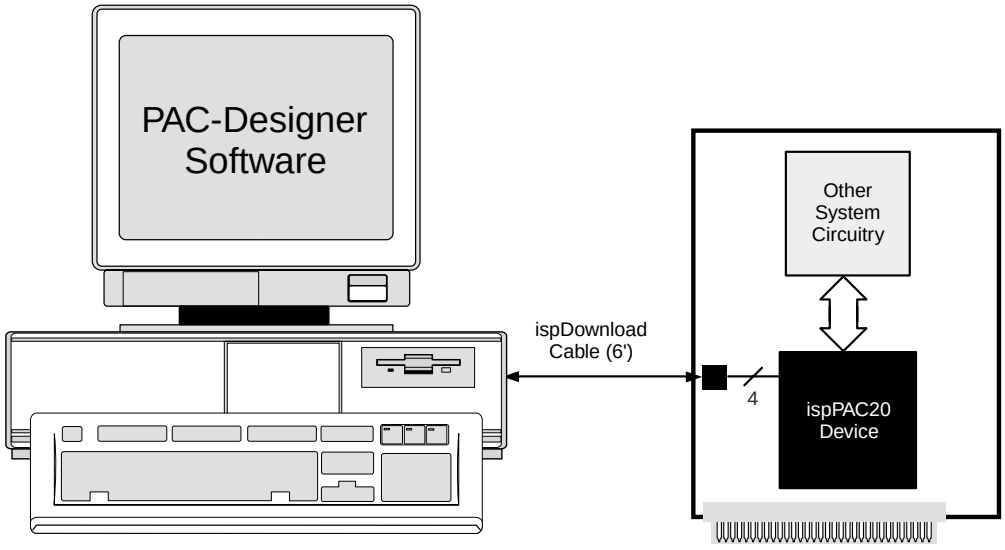
Evaluation Fixture

Included in the basic ispPAC20 Design Kit is an engineering prototype board that is connected to the parallel port of a PC. It demonstrates proper layout techniques for the ispPAC20 and can be used in real time to check circuit operation as part of the design process. Input and output connections as well as a "breadboard" circuit area are provided to speed debugging of the circuit.

User Electronic Signature

A user electronic signature (UES) feature is included in the E^2 memory of the ispPAC20. It contains seven bits that can be configured by the user to store unique data such as ID codes, revision numbers or inventory control data.

Figure 15. Configuring the ispPAC20 "In-System" from a PC Parallel Port



IEEE Standard 1149.1 Interface

Serial Port Programming Interface

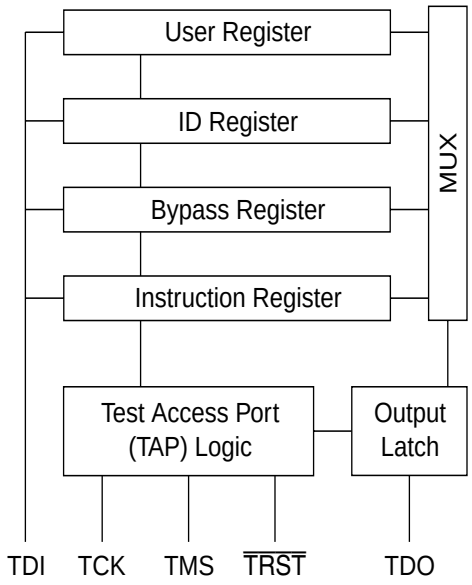
Communication with the ispPAC20 is facilitated via an IEEE 1149.1 test access port (TAP). It is used by the ispPAC20 as a serial programming interface, and not for boundary scan test purposes. There are no boundary scan logic cells in the ispPAC20 architecture. This does not prevent the ispPAC20 from functioning correctly, however, when placed in a valid serial chain with other IEEE 1149.1 compliant devices.

A brief description of the ispPAC20 serial interface follows. For complete details of the reference specification, refer to the publication, Standard Test Access Port and Boundary-Scan Architecture, IEEE Std 1149.1-1990 (which now includes IEEE Std 1149.1a-1993).

Overview

An IEEE 1149.1 test access port (TAP) provides the control interface for serially accessing the digital I/O of the ispPAC20. The TAP controller is a state machine driven with mode and clock inputs. Under the correct protocol, instructions are shifted into an instruction register which then determines subsequent data input, data output, and related operations. Device programming is performed by addressing the user register, shifting data in, and then executing a program user instruction, after which the data is transferred to internal E^2CMOS cells. It is these non-volatile cells that determine the configuration of the ispPAC20. By cycling the TAP controller through the necessary states, data can also be shifted out of the user register to verify the current ispPAC20 configuration. Instructions exist to access all data registers and perform internal control operations.

Figure 16. ispPAC20 TAP Registers



For compatibility between compliant devices, two data registers are mandated by the IEEE 1149.1 specification. Others are functionally specified, but inclusion is strictly optional. Finally, there are provisions for optional data registers defined by the manufacturer. The two required registers are the bypass and boundary-scan registers. For ispPAC20, the bypass register is a 1-bit shift register that provides a short path through the device when boundary testing or other operations are not being performed. The ispPAC20, as mentioned, has no boundary-scan logic and therefore no boundary scan register. All instructions relating to boundary scan operations place the ispPAC20 in the BYPASS mode to maintain compliance with the specification. The optional identification register described in IEEE 1149.1 is also included in the ispPAC20. One additional data register included in the TAP of the ispPAC20 is the Lattice-defined user register. Figure 16 shows how the instruction and various data registers are placed in an ispPAC20.

TAP Controller Specifics

The TAP is controlled by the Test Clock (TCK) and Test Mode Select (TMS) inputs. These inputs determine whether an Instruction Register or Data Register operation is performed. Driven by the TCK input, the TAP consists of a small 16-state controller design. In a given state, the controller responds according to the level on the TMS input as shown in Figure 17. Test Data In (TDI) and TMS are latched on the rising edge of TCK, with Test Data Out (TDO) becoming valid on the falling edge of TCK. There are six steady states within the controller: Test-Logic-Reset, Run-Test/Idle, Shift-Data-Register, Pause-Data-Register, Shift-Instruction-Register, and Pause-Instruction-Register. But there is only one steady state for the condition when TMS is set high: the Test-Logic-Reset state. This allows a reset of the test logic within five TCKs or less by keeping the TMS input high. Test-Logic-Reset is the power-on default state.

When the correct logic sequence is applied to the TMS and TCK inputs, the TAP will exit the Test-Logic-Reset state and move to the desired state. The next state after Test-Logic-Reset is Run-Test/Idle. Until a data or instruction scan is performed, no action will occur in Run-Test/Idle (steady state = idle). After Run-Test/Idle, either a data or instruction scan is performed. The states of the Data and Instruction Register blocks are identical to each other differing only in their entry points. When either block

IEEE Standard 1149.1 Interface

is entered, the first action is a capture operation. For the Data Registers, the Capture-DR state is very simple: it captures (parallel loads) data onto the selected serial data path (previously chosen with the appropriate instruction). For the Instruction Register, the Capture-IR state will always load the IDCODE instruction. It will always enable the ID Register for readout if no other instruction is loaded prior to a Shift-DR operation. This, in conjunction with mandated bit codes, allows a "blind" interrogation of any device in a compliant IEEE 1149.1 serial chain.

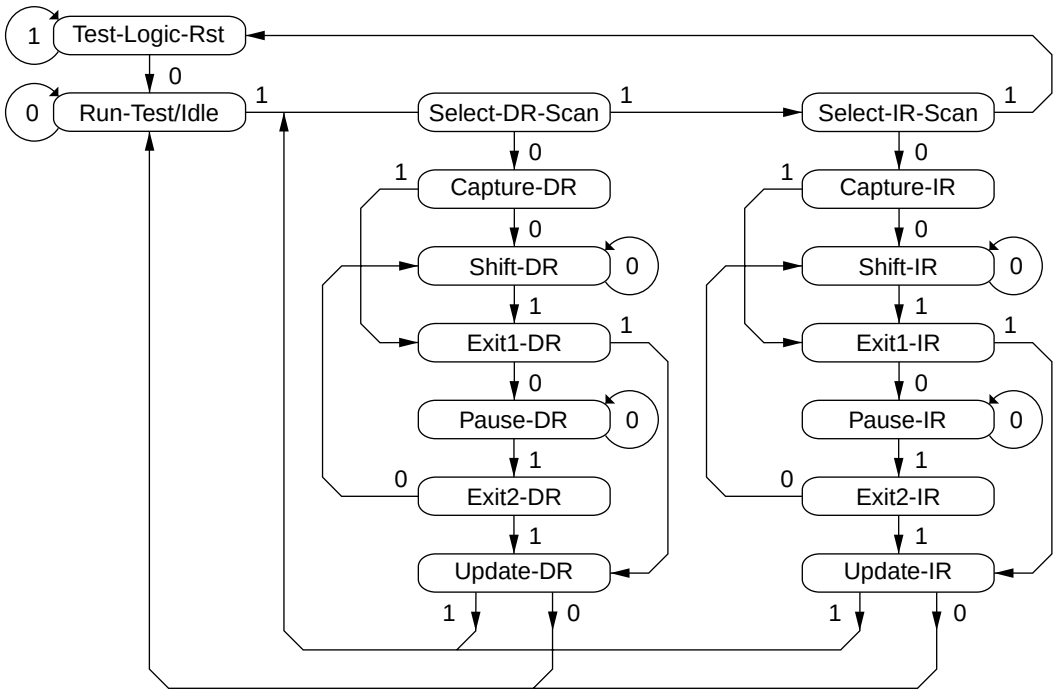
From the Capture state, the TAP transitions to either the Shift or Exit1 state. Normally the Shift state follows the Capture state so that test data or status information can be shifted out or new data shifted in. Following the Shift state, the TAP either returns to the Run-Test/Idle state via the Exit1 and Update states or enters the Pause state via Exit1. The Pause state is used to temporarily suspend the shifting of data through either the Data or Instruction Register while an external operation is performed. From the Pause state, shifting can resume by reentering the Shift state via the Exit2 state or be terminated by entering the Run-Test/Idle state via the Exit2 and Update states. If the proper instruction is shifted in during a Shift-IR operation, the next entry into Run-Test/Idle initiates the test mode (steady state = test). This is when the device is actually programmed, erased or verified. All other instructions are executed in the Update state.

Test Instructions

Like data registers, the IEEE 1149.1 standard also mandates the inclusion of certain instructions. It outlines the function of three required and six optional instructions. Any additional instructions are left exclusively for the manufacturer to determine. The instruction word length is not mandated other than to be a minimum of 2-bits, with only the BYPASS and EXTEST instruction code patterns being specifically called out (all ones and all zeroes respectively). The ispPAC20 contains the required minimum instruction set as well as one from the optional instruction set. In addition, there are several proprietary instructions that allow the device to be configured and verified. For ispPAC20, the instruction word length is 5-bits. All ispPAC20 instructions available to users are shown in Table 6.

BYPASS is one of the three required instructions. It selects the Bypass Register to be connected between TDI and TDO and allows serial data to be transferred through the device without affecting the operation of the

Figure 17. Test Access Port (TAP) Contoller State Diagram



Note: The value shown adjacent to each state transition in this figure represents the signal present at TMS at the time of a rising edge at TCK.

IEEE Standard 1149.1 Interface

Table 6. ispPAC20 TAP Instructions

Instruction	Code	Description
EXTEST	00000	External test. Default to BYPASS.
ADDUSR	00001	Address user data register.
UBE	00010	User bulk erase.
VERUSR	00011	Verify user data register.
PRGUSR	00100	Program user data register.
IDCODE	01101	Read identification data register.
ENCAL	10000	Enable calibration sequence.
DBE	10001	DAC bulk erase.
VERDAC	10010	Verify the DAC register.
PRGDAC	10011	Program the DAC register.
ADDDAC	10100	Address the DAC register.
SAMPLE	11110	Sample/Preload. Default to BYPASS.
BYPASS	11111	Bypass (connect TDI to TDO).

TAP Inst/PAC20

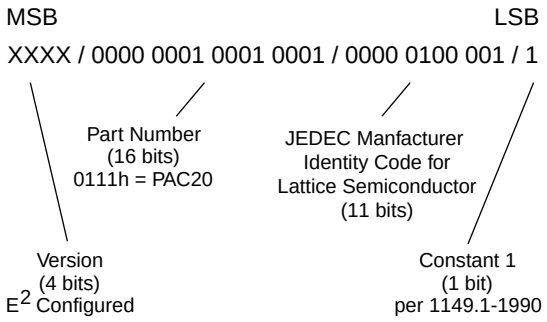
ispPAC20. The bit code of this instruction is defined to be all ones by the IEEE 1149.1 standard.

The required ***SAMPLE/PRELOAD*** instruction dictates the Boundary-Scan Register be connected between TDI and TDO. The ispPAC20 has no boundary-scan register, so for compatibility it defaults to the BYPASS mode whenever this instruction is received. The bit code for this instruction is defined by Lattice as shown in Table 6.

The ***EXTEST*** (external test) instruction is required and would normally place the device into an external boundary test mode while also enabling the Boundary-Scan Register to be connected between TDI and TDO. Again, since the ispPAC20 has no boundary-scan logic, the device is put in the BYPASS mode to ensure specification compatibility. The bit code of this instruction is defined by the 1149.1 standard to be all zeros.

The optional ***IDCODE*** (identification code) instruction is incorporated in the ispPAC20 and leaves it in its functional mode when executed. It selects the Device Identification Register to be connected between TDI and TDO. The Identification Register is a 32-bit shift register containing information regarding the IC manufacturer, device type and version code (see Figure 18). Access to the Identification Register is immediately available, via a TAP data scan operation, after power-up of the device, or by issuing a Test-Logic-Reset instruction. The bit code for this instruction is defined by Lattice as shown in Table 6.

Figure 18. Identification Code (IDCODE) 32-Bit Binary Word for Lattice ispPAC20



ADDUSR (address user register) instruction is a Lattice defined instruction that selects the user register to be shifted during a Shift-DR operation. Normal operation of a device is not interrupted by this instruction. It precedes a PROGUSR (program user) instruction to shift in a new configuration and follows a VERUSR (verify user) instruction to shift out the current configuration. The bit code for this instruction is shown in Table 6.

The ***PRGUSR*** (program user) is a Lattice instruction that enables the data shifted into the user register to be programmed into the non-volatile E^2CMOS memory of the ispPAC20 and thereby alter its configuration. The user register is a 109-bit shift register that contains all the user-controlled parametric and interconnect data pertaining to the configuration of the ispPAC20. Normal operation of the device is interrupted during the actual programming time. A programming operation does not begin until entry of the Run-Test/Idle state. The time required to insure data retention is given in the TAP signal specifications table. The user must ensure that the recommended programming times are observed. The bit code for this instruction is shown in Table 6.

VERUSR (verify user) is the next Lattice instruction and causes the current configuration of the ispPAC20 to be loaded into the user register. This operation doesn't interrupt operation of the device. The current configuration can then be shifted out of the user register immediately after an ADDUSR instruction is executed. The bit code for this instruction is shown in Table 6.

For DAC operations, the ***ADDDAC*** (address DAC), ***PRGDAC*** (program DAC), ***VERDAC*** (verify DAC) and ***DBE*** (DAC bulk erase, instructions are provided. They have basically the same effect as the "user" instructions

IEEE Standard 1149.1 Interface

except that they only affect the contents of the DAC register. The bit codes for these instructions are shown in Table 6.

ENCAL (enable calibration) is a Lattice instruction that enables the start of an auto-calibration sequence. This operation causes all outputs of the device to go to 0V until the calibration sequence is completed (see timing specifications). As with the programming instructions above, calibration does not begin until entry of the Run-Test/Idle state. The completion of the calibration is not dependent, however, on any further TAP control. This means the state of the TAP can be returned immediately to the Test-Logic-Reset state. The only consideration would be to not clock the TAP during critical analog operations. The first several milliseconds of the calibration routine are consumed waiting for configurations to settle, though, leaving more than enough time to clock the TAP back to the Test-Logic-Reset state. The bit code for this instruction is shown in Table 6.

The last Lattice instruction is ***UBE*** (user bulk erase). Operation of the device is interrupted during UBE, after which all inputs are disconnected and all outputs driven to V_{COM} (2.5V). To economize internal circuitry, programming can only be selectively done in one direction (from zeroes to ones). The UBE is used to return all user bits to a zero state at the same time. A UBE usually proceeds a PRGUSR operation, otherwise one to zero changes would not be implemented. It can also be used to erase all configuration information from a device and is the default condition of parts shipped from the factory. The same programming constraints apply to UBE as for PRGUSR. The bit code for this instruction is shown in Table 6.

The ADDUSR, BYPASS, EXTEST, IDCODE and SAMPLE/PRELOAD instructions are all executed in the Update-IR state. Other instructions: PRGUSR, VERUSR and UBE are executed upon entry of the Run-Test/Idle state.

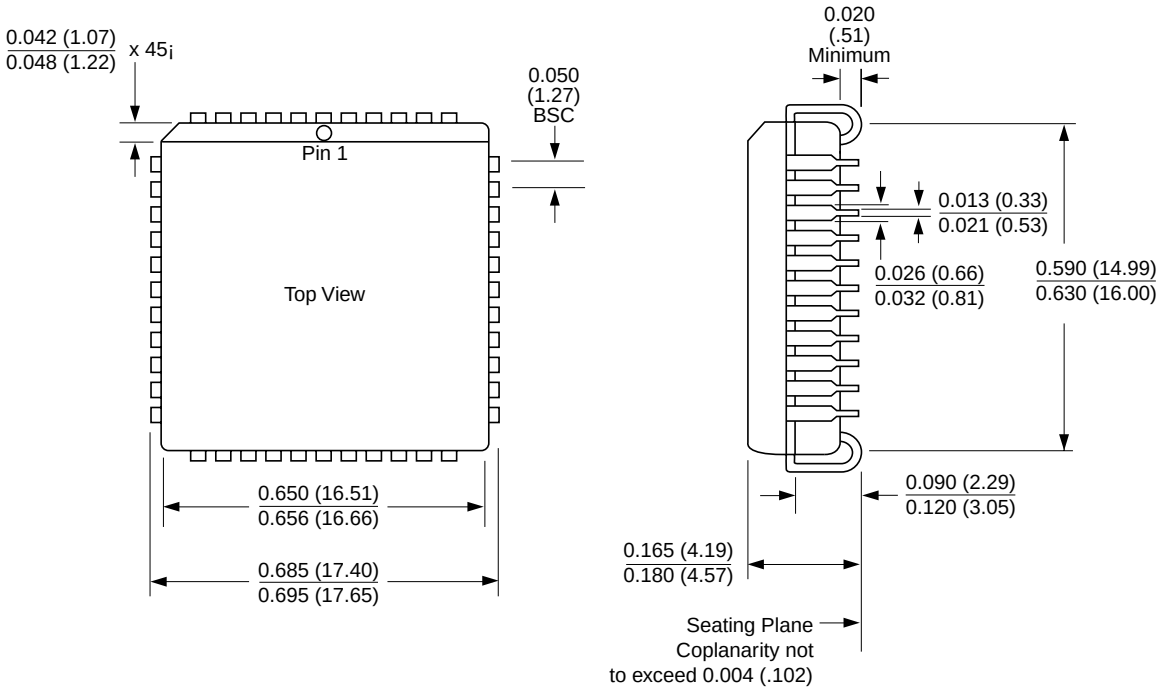
It is recommended that when all serial interface operations are completed, the TAP controller be reset and left in the Test-Logic-Reset state (the power-up default) and the TCK and TMS inputs idled. This will insure the best analog performance possible by minimizing the effects of digital logic “feed-through.”

Package Diagrams

44-Pin Plastic PLCC

Dimensions in Inches MIN./MAX.

(Dimensions in millimeters, shown in parenthesis, are for reference only)



44-Pin TQFP

Dimensions in Millimeters MIN./MAX.