NEMERIX

NJ1030

Rev. 0.9 - Nov. 2003

GPS Baseband Processor

Preliminary Datasheet

1 General Description

The NemeriX NJ1030 is a GPS base-band processor targeting C/A code L1 GPS low power applications and offering WAAS/EGNOS support. It is based on the NemeriX NP1016 GPS correlation core combined with a 32 bit IEE1754 (Sparc v8) compatible CPU core, auxiliary on-chip memory, peripherals and analog blocks. Flexible configuration of system performance and memory architecture allow the NJ1030 to be used as an ultra low power GPS sensor delivering formatted navigation information (NMEA or others) or as a flexible mid range micro controller with GPS capabilities.

The GPS correlation is carried out by the NemeriX NP1016 correlator IP core with 64 correlators (configured as 16 tracking modules with 4 correlators each). While supporting any GPS RF front-end device with 1 bit or 2 bits output and a sampling frequency of up to 20MHz, the NJ1030 is optimized to work with the NemeriX NJ100x RF front ends. In particular the specific power management modes of the NJ100x (NJ1004, NJ1006) are directly controlled. Flexible clocking schemes are implemented, making use of internal or external clock sources.

The CPU is a 32 bit RISC IEEE1754 (SPARC V8) compatible core with 8kBytes instruction cache and 1 kBytes data cache. Moreover 32 kBytes of on chip SRAM are included. A dedicated, battery backed-up, 8kBytes SRAM block for the storage of navigation information and a real time clock can be used to accelerate the space vehicle (SV) acquisition at power up.

External memory and I/O space is accessed through a 32bit external bus interface (EBI), which supports up to 4 banks of SRAM or Flash memory of 16MBytes each. One UART plus optionally a 2 slaves SPI interface, a second UART or a general-purpose interface (GPIO) are available.

An 8 bit analog to digital converter combined with an on chip temperature sensor may eliminate the need for a temperature compensated crystal oscillator (TCXO) in the system. Seven external analog inputs can be used to integrate other sensors (such as a compass or a gyroscope) for enhanced navigation applications.

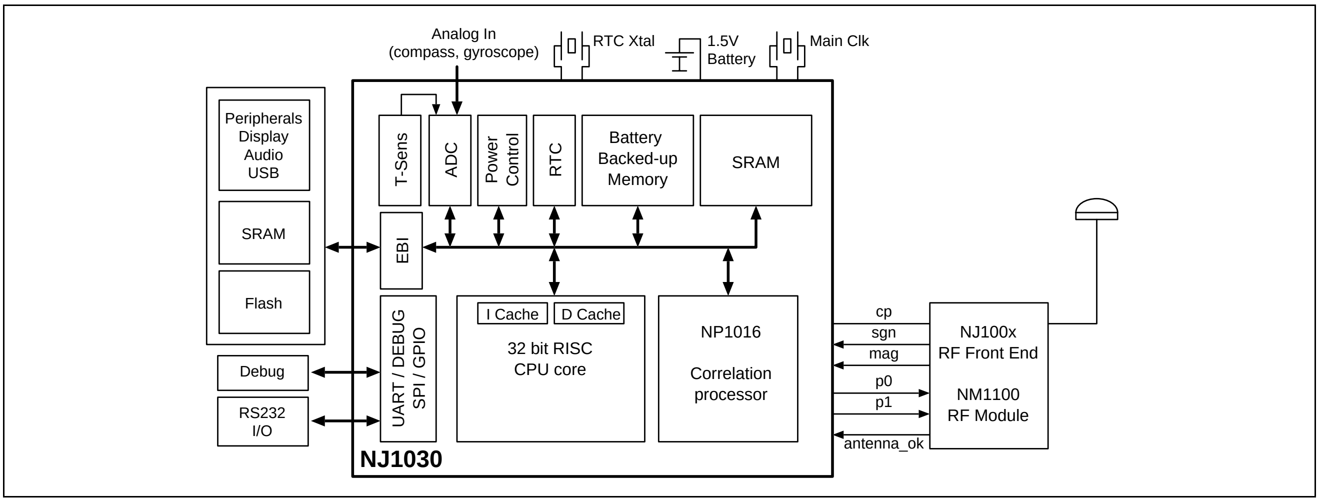
1.1 Features

- Single chip, stand-alone GPS base-band processor
- Low power: less than 25 mW in fully active operation
- Aided GPS compatible (3GPP TS44.035 – TIA-IS801)
- 16 C/A-code tracking modules with 4 correlators each
- WAAS/EGNOS ready
- 32 bit, royalty free, processor (Sparc V8 compatible)
- 32 kBytes of SRAM
- 8 kBytes battery backed up SRAM and real time clock
- UART and a selectable UART/SPI/GPIO interface
- 32/16 or 8 bit external bus interface (EBI)
- 2-bit sign and magnitude GPS IF signal input
- 8 bit ADC: temperature sensor and 7 analog inputs
- Package: micro BGA128 7mm x 7mm, 0.5 mm pitch

1.2 Applications

- Mobile handsets
- Stand-alone, battery-operated GPS receivers
- Automotive navigation systems
- GPS-time wristwatches and time standards

Figure 1 – NJ1030 based GPS receiver



Absolute Maximum Ratings

Max. Supply Voltage (LVDD, VBAT)..........................2.2V
Max. Supply Voltage (all other).................................4.2V
Max. voltage on any pin.......-0.3V to corresp. xVDD+0.3V
Max. current into any pin.......................................±20mA
Continuous Power Dissipation...............................200mW

Operating temperature ....–40 to +85°C
Junction temperature ....125°C
Storage temperature ....–65 to +150°C
Lead temperature (Soldering, 10s)..........................260°C

Absolute maximum ratings are short term stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the recommended operating conditions is not implied. Stresses beyond those listed under absolute maximum ratings may cause permanent damage to the device. Exposure to absolute maximum ratings conditions for extended periods may affect device reliability

ESD sensitive device: use proper precautions when handling this device.

Electrical Characteristics

1.6V ≤ DVDD ≤ 3.6V, 1.6V ≤ AVDD ≤ 3.6V, 1.6V ≤ DVSU ≤ 2V, 1.6V ≤ TVDD ≤ 3.6V, 1.2V ≤ LVDD ≤ 1.8V, VBAT ≤ 2V, T_{amb} = -40º to +85ºC, load = 10 pF, crystal oscillator active, f_{sys_clk}=16.368 MHz. All voltages referred to xGND. Typical values are at xVDD = 2.5V and T_{amb}=+25°C

Parameter	Conditions/description	Min	Typ	Max	Unit	Notes
DC Parameters						
AVDD, TVDD, DVDD	Padring Supply Voltage	1.6		3.6	V	
LVDD	Core Power Supply	1.2		2	V	1
DVSU	Power Up Unit Supply	1.6		2	V	
VBAT	Battery Power Supply	1.1	1.5	2.0	V	2
I_LVDD	Average current with: On-chip LVDD = 1.25V Other xVDD = 1.8V Chip Fully Active	6	12	14	mA	3
I_DVDD			TBD		mA	3
I_TVDD			TBD		mA	3
I_AVDD			TBD		mA	3
Battery OK Flag Threshold	VBAT Rising		1.15		V	
	VBAT Falling		1.10		V	
Power Supervisor						
VREF	Voltage reference	1.22	1.24	1.26	V	
VSI	Voltage Superv. Threshold		VREF		V	
I_HYST	Hysteresis Current	1.1	1.6	2.1	µA	
AC Parameters						
Sys Clock Frequency	LVDD = 1.8V	0		98.2	MHz	
	LVDD = 1.2V	0		32.4	MHz	
Sys Clock Crystal Drive Level			10		µW	
RTC Clock Crystal Drive Level			100		nW	
Clock Input Level	Clipped Sine	100			mV	5
Digital Interfaces						
VIH		0.7 DVDD			V	
VIL				0.3	V	
VOH	Ioh = -50 µA	0.9 DVDD			V	
VOL	Iol = 50 µA			0.1 DVDD	V	
Output Rise Time	Cload = 15 pF			5	ns	
Output Fall Time	Cload = 15 pF			5	ns	

Note 1: Core (LVDD) may be supplied by the internal voltage regulator.
Note 2: Guarantees NVRAM data retention and RTC operation.
Note 3: CPU executing GPS positioning, driving a NJ1006 RF front end with a 16.368 MHz clock
Note 4: Hysteresis current is turned on when VSI>VREF (See section 22.2).
Note 5: X_{osc} used also as TCXO buffer.

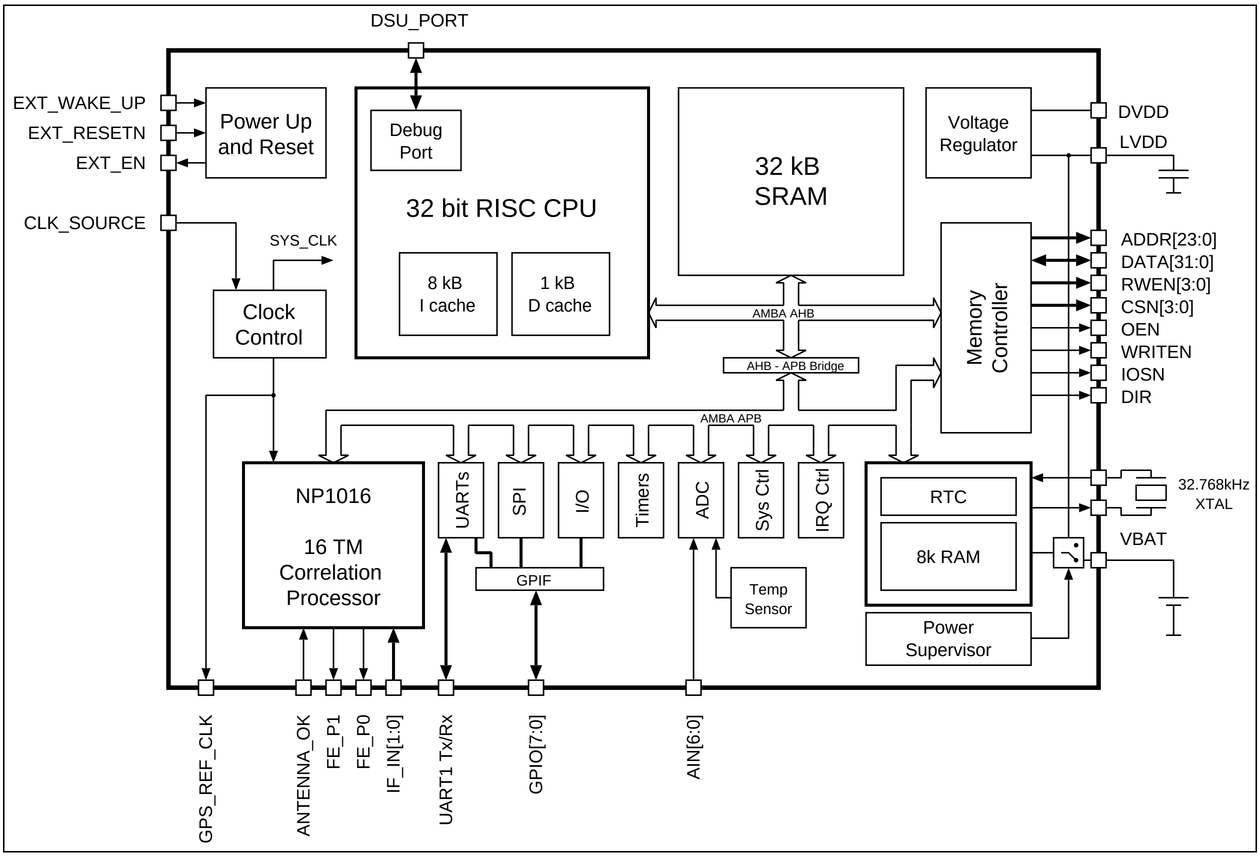
2 GPS Performance

All performance data refer to a NJ1030 used with a NJ1006 RF front end and the antenna used in the DK1030 GPS development kit and running version 0.8 of the NS1030 GPS software.

Parameter	Conditions/description	Min	Typ	Max	Unit	Notes
Time to first fix						
TTFF	Cold Start		43		s	1
	Warm Start		38		s	1
	Hot Start		9		s	1
Sensitivity						
Acquisition	Cold Start		-132		dBm	2
	Warm Start		-133		dBm	2
	Hot Start		-134		dBm	2
Tracking			-147		dBm	
Accuracy	CEP50		3.5		m	3
	CEP90		6.4		m	3
Acceleration				4	g	
Jerk				4	g/s	

Note 1: Outdoor – clear sky situation. At least 4 SV with a signal stonger than -124 dBm.
Note 2: The average TTFF is 2'27 for cold start, 1'03'' for warm and 1'18'' for hot start situation.
Note 3: Measured over 24 hours

Figure 2. NJ1030 Block diagram



3 Architectural Overview

This chapter briefly describes the main building blocks of the NJ1030. A more detailed description of all blocks and their sub-blocks can be found in the following chapters.

3.1 CPU

The NJ1030 CPU is a royalty free, 32bit RISC processor conforming to the IEEE_1754 (SPARC V8) architecture. It is designed for embedded applications and provides the following features:

- 5 stages pipelined architecture
- single cycle 32 bit hardware multiplier
- barrel shifter
- radix 2 hardware divider
- 8 windows register file
- 8 kBytes instruction cache, 1 kBytes data cache
- AMBA AHB and APB compatible interface
- Debug port (DSU)

3.2 On Chip SRAM

A block of 32kBytes 0-wait states SRAM is available on chip, on the fast AHB bus. This allows critical code portions as well as data structures to be stored. The on-chip SRAM content is cachable (both instructions and data). This memory supports 8,16 and 32 bit access mode.

3.3 Battery Backed up Memory

In order for the software to store recent GPS almanac data at boot time, the NJ1030 has a block of 8 kBytes of battery backed-up and write protected SRAM (NVRAM). The power supply of this memory block is monitored by the integrated power supervisor. This memory area can also be used to store relevant system status information before entering power management modes that involve the removal of the power supply. Only 32 bit access is supported.

3.4 External Bus Interface (EBI)

The EBI directly supports up to 4 banks of 16 MBytes of asynchronous memory, with a fifth bank optionally accessible through signals of the GPIO interface. Each bank can individually be set to implement an 8,16 or 32 bit access mode with a selectable number of wait states.

3.5 Clocking

The main system clock can be generated by an on chip oscillator, by a TCXO or be an external digital clock. This is the clock used for all the NJ1030 functions, excluding the GPS correlation that has a dedicated clock derived internally from the system clock (with a division factor between 1 to 8). This allows for flexible frequency plans to be implemented, with the possibility to trade off between the computational requirements of the application, the GPS frequency plan and the power consumption requirements.

3.6 GPS Correlation Unit

The GPS correlation function is carried out by the correlation unit which is a 16 tracking modules (64 correlators) implementation of the NemeriX NP1016 GPS correlator IP core. The NP1016 is connected to the CPU as an AMBA APB peripheral. Two GPS interrupt signal (ACC_INT and MEAS_INT) are connected to the CPU interrupt controller. The correlation block is clocked by a dedicated clock (NP1016_GPS_CLK) signal derived from the main SYS_CLK, which can be deactivated when the GPS functionality is not needed by the system.

The correlation unit provides:

- 16 C/A-code acquisition and tracking modules (TMx)
- WAAS/EGNOS support
- 2-bit sign and magnitude signal input
- Programmable interrupt interval and measurement rate
- 1 pulse-per-second (1PPS) GPS time reference
- Individual tracking-module activation
- Power management modes
- Direct control of the NJ100x RF front end power management features
- Fast acquisition mode (FAM)

3.7 Other Peripherals

- **UARTs** : two serial interfaces are available. UART1 is always available on dedicated pins, while a second UART shares the pins of the GPIO interface. UART1 has a 16 bytes receive and send FIFO, while UART 2 has a 4 bytes send and receive FIFO. The UART1 lines can also be configured to act as DSU UART lines, allowing for flash re-programming reusing the buffer and connectors of UART1, without the need to access the dedicated DSU pins.
- **SPI** : both master and slave SPI interface alternatively are supported, by sharing the same pins of the GPIO interface. The SPI master supports up to 2 slaves. The slave has a send and receive buffer of 16 bytes.
- **Timers**: two 24-bit timers are provided on-chip. The timers can work in periodic or one-shot mode. Both timers are clocked by a common 10-bit prescaler.
- **Watchdog** : a 24-bit watchdog is provided on-chip. The watchdog is clocked by the timer prescaler. When the watchdog reaches zero, an output signal (WDOG) is asserted. If enabled, this signal can be used to generate system reset.
- **Interrupt controller** : it manages a total of 15 interrupts, originating from internal and external sources. Each interrupt can be programmed with a two levels priority. Internal and external interrupt sources are used.
- **Analog to Digital Converter (ADC)** : an 8 bit resolution 8 channels ADC is available with 7 analog inputs directly connected to the pads and one input connected to an on chip temperature sensor.
- **RTC**: a pseudo real time clock block with 1 second precision, 30 bits wide register is implemented. It also offers a wake up functionality that can be used to recover the system from power down modes. The RTC has an on chip oscillator that uses a 32 kHz quartz.

3.8 GPIO Interface

An 8 bit general purpose I/O interface allows for parallel I/O. This interface can be reconfigured in order to share some lines with other peripherals.

3.9 GPS IF Interface

This dedicated interface consist of 2 bits input (sign and magnitude), two power mode control signals for the NJ100x RF front ends, the GPS clock and an antenna ok input signal. The GPS IF interface also has a dedicated power supply (TVDD).

3.10 Power Supplies

Different power supplies for pad ring and core are used, with the possibility to generate the core power supply by an on chip voltage regulator. The core voltage can be adjusted between 1.2V and 1.8V, depending on the clock frequency being used. The on chip voltage regulator can also be disabled if an external power supply is available in the system.

The digital pad ring power supply can be between 1.8V and 3.3V, with the possibility also to use an independent level for the IF interface. In power down mode a third digital power supply is used to maintain a tiny portion of the logic ready to wake up the NJ1030 and the attached components.

For the battery backed up memory and the RTC a dedicated power supply is used. The power supervisor can be programmed to monitor the used core or padring power supply. In case of faillure of the monitored power supply, the power supervisor will switch the power supply source to battery for the RTC and battery backed up memory as soon as the programmed threshold is reached.

3.11 Power Down Modes

Beside fully active operation, where individual peripherals are clocked only depending on their activity, the NJ1030 supports STAND-BY and SLEEP modes.

The STAND-BY mode is entered by the CPU when no other instruction has to be executed, and only an interrupt can resume software execution. In this status, the clocking activity is reduced to the minimum allowed by the application, with the CPU basically in a wait status.

The SLEEP mode is entered by the CPU when a period of complete system inactivity can be expected. At this point the CPU can turn off the core power supply as well as other external components. Only a tiny portion of logic is kept powered and will reboot the system at the occurrence of a RTC wake up or an external wake up or reset.

4 Pinout Table

Pin Nb	Ball Nb	Signal	Type	Description	LVDD=0 status
1	C4	DATA[4]	Digital IO	EBI Data Bus	Z + keeper
2	C3	DATA[5]	Digital IO	EBI Data Bus	Z + keeper
3	C2	DATA[6]	Digital IO	EBI Data Bus	Z + keeper
4	C1	DATA[7]	Digital IO	EBI Data Bus	Z + keeper
5	D4	DATA[8]	Digital IO	EBI Data Bus	Z + keeper
6	D3	DATA[9]	Digital IO	EBI Data Bus	Z + keeper
7	D2	DATA[10]	Digital IO	EBI Data Bus	Z + keeper
8	D1	DATA[11]	Digital IO	EBI Data Bus	Z + keeper
9	E4	DATA[12]	Digital IO	EBI Data Bus	Z + keeper
10	E3	DATA[13]	Digital IO	EBI Data Bus	Z + keeper
11	E2	DATA[14]	Digital IO	EBI Data Bus	Z + keeper
12	E1	DATA[15]	Digital IO	EBI Data Bus	Z + keeper
13	F2	DATA[16]	Digital IO	EBI Data Bus	Z + keeper
14	F3	DATA[17]	Digital IO	EBI Data Bus	Z + keeper
15	F1	LVDD	Digital power	Core power supply	
16	F4	LVSS	Digital ground	Ground	
17	G1	DATA[18]	Digital IO	EBI Data Bus	Z + keeper
18	G2	DATA[19]	Digital IO	EBI Data Bus	Z + keeper
19	G3	DATA[20]	Digital IO	EBI Data Bus	Z + keeper
20	H1	DATA[21]	Digital IO	EBI Data Bus	Z + keeper
21	G4	DVDD	Digital power	Power supply for padring and voltage reg.	
22	H4	DVSS	Digital gnd	Ground	
23	H3	DATA[22]	Digital IO	EBI Data Bus	Z + keeper
24	H2	DATA[23]	Digital IO	EBI Data Bus	Z + keeper
25	J1	DATA[24]	Digital IO	EBI Data Bus	Z + keeper
26	J2	DATA[25]	Digital IO	EBI Data Bus	Z + keeper
27	J3	DATA[26]	Digital IO	EBI Data Bus	Z + keeper
28	J4	DATA[27]	Digital IO	EBI Data Bus	Z + keeper
29	K1	DATA[28]	Digital IO	EBI Data Bus	Z + keeper
30	K2	DATA[29]	Digital IO	EBI Data Bus	Z + keeper
31	K3	DATA[30]	Digital IO	EBI Data Bus	Z + keeper
32	K4	DATA[31]	Digital IO	EBI Data Bus	Z + keeper
33	L1	DIR	Digital Out	EBI Data Bus direction	0
34	M1	OEN	Digital Out	EBI Output Enable	1
35	M2	IOSN	Digital Out	EBI I/O Space chip select	1
36	L2	WRITEN	Digital Out	EBI I/O Space Write Enable	1
37	L3	RWEN[0]	Digital Out	EBI RAM Write Strobe / Byte Select	1
38	M3	RWEN[1]	Digital Out	EBI RAM Write Strobe / Byte Select	1
39	L4	RWEN[2]	Digital Out	EBI RAM Write Strobe / Byte Select	1
40	M4	RWEN[3]	Digital Out	EBI RAM Write Strobe / Byte Select	1
41	J5	CSN4	Digital Out	EBI RAM4 Chip Select	1
42	K5	CSN0	Digital Out	EBI RAM0 Chip Select (Boot)	1
43	L5	CSN1	Digital Out	EBI RAM1 Chip Select	1
44	M5	CSN2	Digital Out	EBI RAM2 Chip Select	1
45	J6	CLK_SOURCE[1]	Digital In	External/Internal clock selector	
46	K6	EXT_WAKE_UP[1]	Digital In	External Wake Up Signal	
47	L6	EXT_RESETN[1]	Digital In	Active Low Asynchronous Reset	
48	M6	EXT_EN[1]	Digital Out	Enable for ext components (osc. voltage reg.)	0
49	L7	EXT_VREGN[1]	Digital In	Internal/External voltage regulator selector	
50	K7	DVSU	Digital power	Independent power supply for Power-up unit	
51	M7	TVDD	Digital power	RF interface power supply	
52	J7	TVSS	Digital ground	RF interface ground	

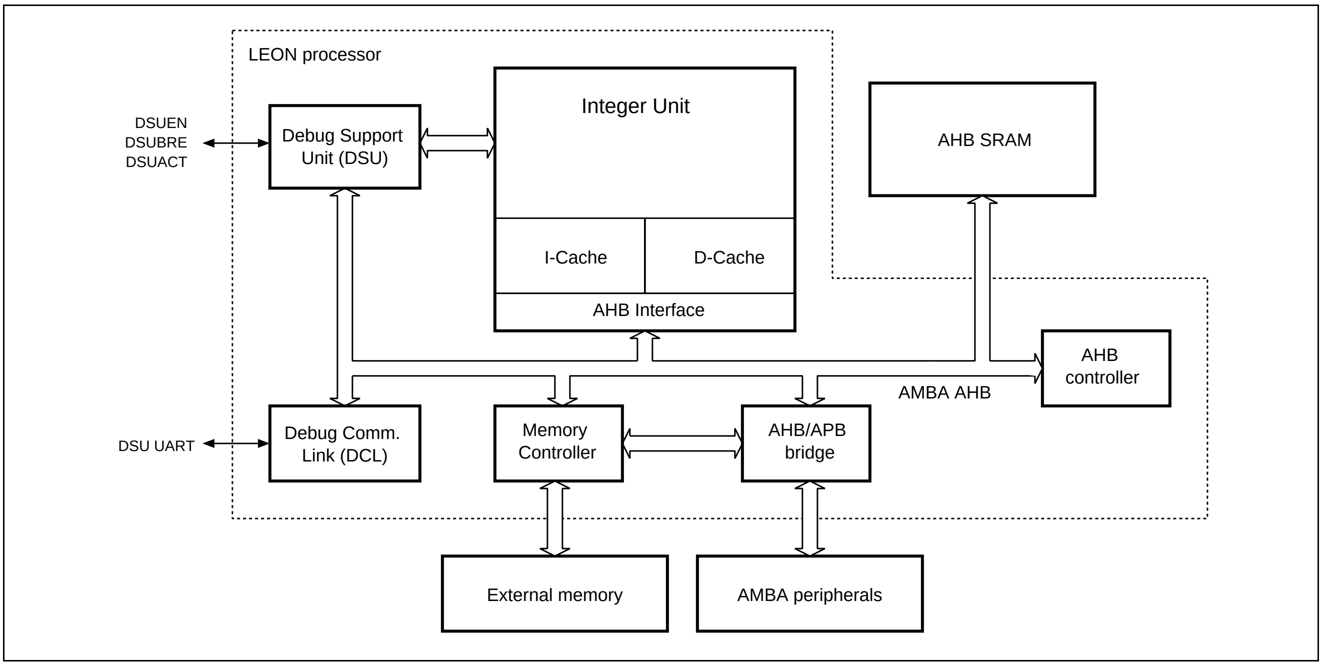
53	M8	GPS_REF_CLK[2]	Digital Out	GPS reference clock output	
54	J8	ANTENNA_OK[2]	Digital In	Antenna status indicator	
55	L8	MAG[2]	Digital In	GPS IF input signal (magnitude)	
56	K8	SGN[2]	Digital In	GPS IF input signal (sign)	
57	K9	FE_P0[2]	Digital Out	Power control signal to RF	0
58	J9	FE_P1[2]	Digital Out	Power control signal to RF	0
59	L9	VSI[3]	Analog In	Power supervisor analog input	
60	M9	AVDD	Analog power	Analog power supply	
61	L10	AVSS	Analog gnd	Analog ground	
62	M10	VREF[3]	Analog Out	Voltage reference	
63	M11	MXI[3]	Analog/Dig. In	Main Crystal or external clock input	
64	M12	MXO[3]	Analog Out	Main Crystal Out	
65	J10	AIN[3][3]	Analog In	Analog signal input to ADC	
66	K10	AIN[1][3]	Analog In	Analog signal input to ADC	
67	L11	AIN[5][3]	Analog In	Analog signal input to ADC	
68	K11	AIN[0][3]	Analog In	Analog signal input to ADC	
69	J11	AIN[4][3]	Analog In	Analog signal input to ADC	
70	L12	AIN[2][3]	Analog In	Analog signal input to ADC	
71	K12	AIN[6][3]	Analog In	Analog signal input to ADC	
72	H10	AVSS	Analog ground	Analog ground	
73	J12	RXI	Analog In	RTC Crystal Input	
74	H12	RXO	Analog Out	RTC Crystal Out	
75	H11	VBAT	Analog power	Back-up battery	
76	H9	TEST_MODE[1]	Digital In	Reserved	
77	G11	TEST_MODE[0]	Digital In	Reserved	
78	G10	DSU_MUX	Digital In	UART1/DSU select	
79	G9	LVSS	Digital ground	Ground	
80	G12	LVDD	Digital power	Core power supply	
81	F12	VFB	Analog In	Voltage regulator feedback	
82	F11	DSUACT	Digital Out	Debug Support Unit (Active)	1
83	F10	DSUEN	Digital In	Debug Support Unit (Enable)	
84	F9	DSURX	Digital In	Debug Support Unit (Receive)	
85	E10	DSUBRE	Digital In	Debug Support Unit (Break)	
86	E11	DSUTX	Digital Out	Debug Support Unit (Transmit)	1
87	E9	DVSS	Digital ground	Ground	
88	D9	DVDD	Digital power	Padring power supply	
89	E12	GPIO[0]	Digital IO	Programmable UART2/SPI/GPIO interface	Z + keeper
90	D12	GPIO[1]	Digital IO	Programmable UART2/SPI/GPIO interface	Z + keeper
91	D11	GPIO[2]	Digital IO	Programmable UART2/SPI/GPIO interface	Z + keeper
92	D10	GPIO[3]	Digital IO	Programmable UART2/SPI/GPIO interface	Z + keeper
93	C11	GPIO[4]	Digital IO	Programmable UART2/SPI/GPIO interface	Z + keeper
94	C10	GPIO[5]	Digital IO	Programmable UART2/SPI/GPIO interface	Z + keeper
95	C12	GPIO[6]	Digital IO	Programmable UART2/SPI/GPIO interface	Z + keeper
96	B11	GPIO[7]	Digital IO	Programmable UART2/SPI/GPIO interface	Z + keeper
97	B12	UART1_RX	Digital In	UART1 Receive	
98	A12	UART1_TX	Digital Out	UART1 Transmit	
99	C9	ADDR[23]	Digital Out	EBI Address	1
100	A11	ADDR[22]	Digital Out	EBI Address	1
101	B10	ADDR[21]	Digital Out	EBI Address	1
102	A10	ADDR[20]	Digital Out	EBI Address	1
103	B9	ADDR[19]	Digital Out	EBI Address	1
104	A9	ADDR[18]	Digital Out	EBI Address	1
105	D8	ADDR[17]	Digital Out	EBI Address	1
106	C8	ADDR[16]	Digital Out	EBI Address	1

107	B8	ADDR[15]	Digital Out	EBI Address	1
108	A8	ADDR[14]	Digital Out	EBI Address	1
109	B7	ADDR[13]	Digital Out	EBI Address	1
110	C7	ADDR[12]	Digital Out	EBI Address	1
111	D7	DVSS	Digital ground	Ground	
112	D6	DVDD	Digital power	Power supply for padring and voltage reg.	
113	A7	ADDR[11]	Digital Out	EBI Address	1
114	B6	ADDR[10]	Digital Out	EBI Address	1
115	C6	ADDR[9]	Digital Out	EBI Address	1
116	A6	ADDR[8]	Digital Out	EBI Address	1
117	A5	ADDR[7]	Digital Out	EBI Address	1
118	B5	ADDR[6]	Digital Out	EBI Address	1
119	C5	ADDR[5]	Digital Out	EBI Address	1
120	D5	ADDR[4]	Digital Out	EBI Address	1
121	A4	ADDR[3]	Digital Out	EBI Address	1
122	A3	ADDR[2]	Digital Out	EBI Address	1
123	A2	ADDR[1]	Digital Out	EBI Address	1
124	A1	ADDR[0]	Digital Out	EBI Address	1
125	B4	DATA[0]	Digital IO	EBI Data Bus	Z + keeper
126	B3	DATA[1]	Digital IO	EBI Data Bus	Z + keeper
127	B2	DATA[2]	Digital IO	EBI Data Bus	Z + keeper
128	B1	DATA[3]	Digital IO	EBI Data Bus	Z + keeper

Note 1: Pads under DVSU power supply.
Note 2: Pads under TVDD power supply.
Note 3: Pads under AVDD power supply.

5 CPU

Figure 3. Leon block diagram



The following section gives a general overview of the Leon's building blocks. More details can be found in the next chapters or in the online Leon and Sparc V8 documentation available at http://www.gaisler.com. The Leon version implemented in the NJ1030 design is 1.0.16.

5.1 Integer Unit

The LEON integer unit implements the full SPARC V8 instruction set, including all multiply and divide instructions. The number of register windows used in the NJ1030 is 8 (see Sparc V8 reference).

5.2 Cache Sub-System

The Leon is configured with 8 kBytes of direct mapped instruction cache and 1 kByte of direct mapped data cache. The instruction cache uses streaming during line-refill to minimise refill latency. The data cache uses write-through policy and implements a double-word write-buffer.

5.3 Debug Support Unit

The debug support unit (DSU) allows debugging on target hardware and has negligible impact on performance. The DSU allows inserting breakpoints and access to all on-chip registers from a remote debugger. Communication to an outside debugger (e.g. gdb) is done using a dedicated UART (RS232).

5.4 Watchpoint Registers

To aid software debugging, four watchpoints registers are available. Each register can cause a trap on an arbitrary instruction or data address range. If the debug support unit is enabled, the watchpoints can be used to enter debug mode.

5.5 Memory Map

Address Space	Size	Mapping	Chip Select	Cacheable
0x00000000 - 0x00FFFFFF	16 M	RAM0 (boot)	CSN0	I and D
0x01000000 - 0x01FFFFFF	16 M	RAM1	CSN1	I and D
0x02000000 - 0x02FFFFFF	16 M	RAM2	CSN2	I and D
0x03000000 - 0x03FFFFFF	16 M	RAM3	CSN3[1]	I and D
0x04000000 - 0x0FFFFFFF	192 M	Unused	Unused	I and D
0x10000000 - 0x10FFFFFF	16 M	RAM4	CSN4	I and D
0x11000000 - 0x1FFFFFFF	240 M	Unused	Unused	I and D
0x20000000 - 0x20FFFFFF	16 M	IO0	IOSN	no
0x21000000 - 0x2FFFFFFF	16 M	IO0 (echo)	IOSN	no
0x30000000 - 0x6FFFFFFF	16 M	IO0 (echo)	IOSN	I only
0x70000000 - 0x7FFFFFFF	32 K	AHBSRAM	-	I and D
0x80000000 - 0x8FFFFFFF	256 M	APB area	-	-
0x90000000 - 0x9FFFFFFF	256 M	Debug Support Unit	-	-

Note 1: Available through GPIO interface.

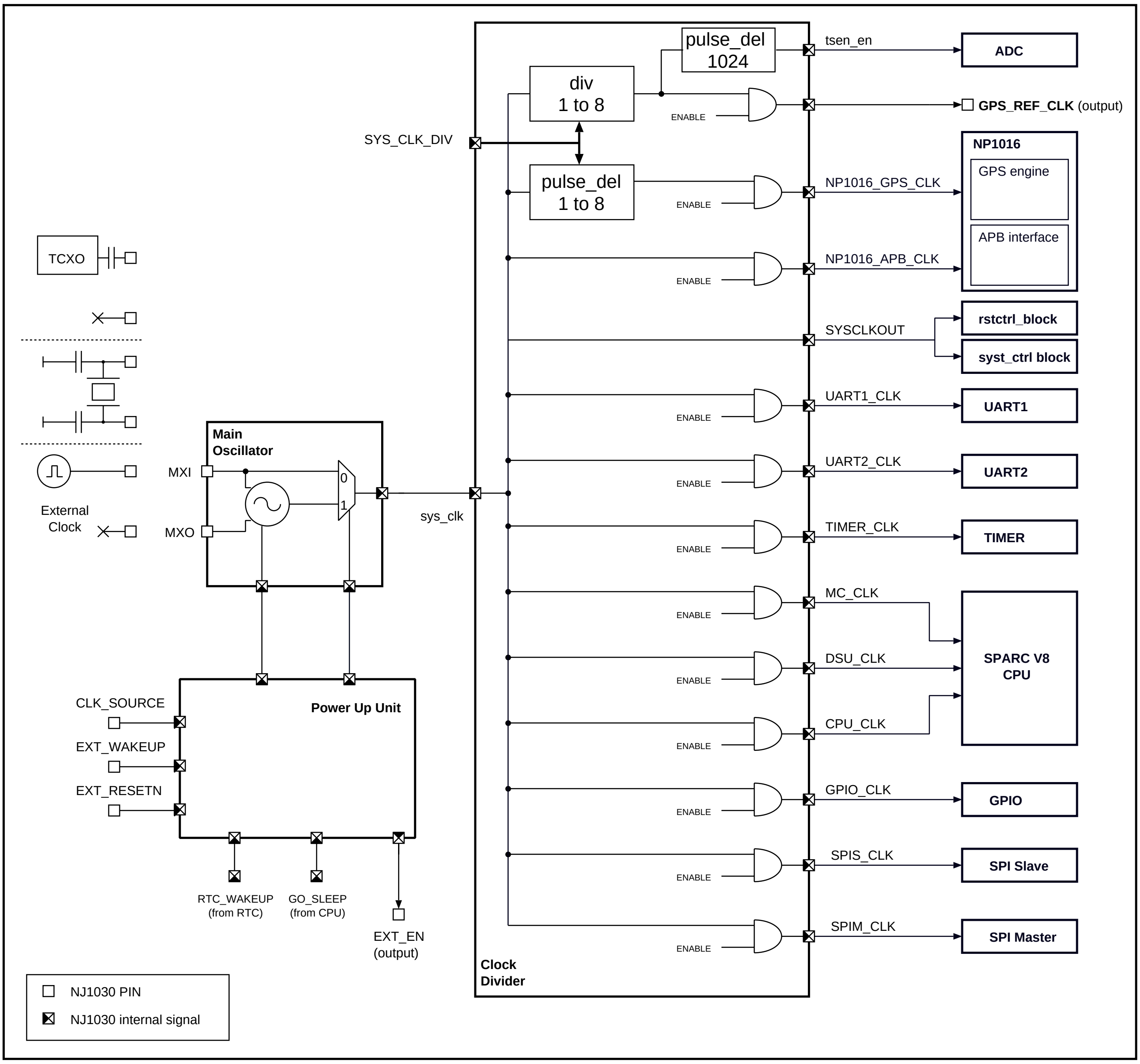
5.6 APB Register Mapping

Slave Id	Function	Internal Amba bus address
0	Memory controller	0x80000000 - 0x80000008
1	AHB status reg	0x8000000C - 0x80000010
2	Cache controller	0x80000014 - 0x80000014
3	Write protection	0x8000001C - 0x80000020
4	Configuration register	0x80000024 - 0x80000024
5	Timers	0x80000040 - 0x8000005C
6	UART 1	0x80000070 - 0x8000007C
7	UART 2	0x80000080 - 0x8000008C
8	Interrupt CTRL	0x80000090 - 0x8000009C
9	I/O Port	0x800000A0 - 0x800000AC
10	Temperature sensor	0x800000B0 - 0x800000BC
11	DSU UART	0x800000C0 - 0x800000CC
12	SPI Master	0x800000D0 - 0x800000DC
13	SPI Slave	0x800000E0 - 0x800000E8
14	System CTRL Block	0x800000F0 - 0x800000FC
15	NP1016	0x80001000 - 0x80001FFC
16	NVRAM RTC[1]	0x80002000 - 0x80003FFC 0x80008000 - 0x8000800C

Note 1 : The NVRAM memory and the RTC module share the same APB slave peripheral.
Note 2 : For each function, the address range is intended from the first to the last register included.

6 Clocking

Figure 4. Clocking scheme



The main clock (SYS_CLK) can be generated either by:

- the on-chip crystal oscillator (Xosc).
- a temperature compensated oscillator (TCXO).
- an external digital clock.
- the RF front-end oscillator (full swing).

All along the datasheet, the following rule has been adopted:

An *'internally generated clock'* refers to a Xosc or a TCXO.

An *'externally generated clock'* refers to any clock source that can be directly used by the NJ1030 without being buffered (external digital clock or clock generated by the RF front-end)

The clocking of the system is flexible enough to scale the system performance depending on the application, to support a wide range of GPS frequency plans and to allow the use of different clock sources. The NJ1030 has two synchronous clock domains:

- SYS_CLK: used for the CPU and all the peripherals except the NP1016. It can run at up to 98.2 MHz.
- GPS_REF_CLK/NP1016_GPS_CLK: the reference clock for the RF front-end and the correlation unit. It is generated from SYS_CLK by means of a programmable clock divider under control of the NJ1030 CPU. The clock division factor range is from 1 to 8.

For the implementation of the power management modes, the SYS_CLK generation can be switched off by the CPU. The system clocking is then resumed either by an external wake up, by a system reset or by the internal RTC alarm. The CLK_SOURCE pin selects between an external ('0') or internal ('1') clock source.

6.1 Shut Down of Main Clock in SLEEP Mode

If the on chip oscillator is used, then it is automatically de-activated when the system enters SLEEP mode.

When the system enters SLEEP mode, the internal oscillator is always selected as the source of the clock (independently from the value of CLK_SOURCE). This feature effectively shuts down the NJ1030 main clock even if the external clock is still running.

The EXT_EN pin can be used to notify an external generator that the NJ1030 has entered sleep mode (see chapter 23, Note on EXT_EN functionality)

6.2 Sub-Clocks

A set of subclocks is generated from the main source. All the following subclocks are either divided version or gated version of the main clock.

- **GPS_REF_CLK**: GPS reference clock available on the external pin for the GPS RF front end. It is a divided version of SYS_CLK. Clock division factor is 1 to 8 according to the value of CLK_DIV_FACTOR.
- **NP1016_GPS_CLK**: pulse deleted version of SYS_CLK. Pulse deletion factor is between 1 and 8 according to the value of CLK_DIV_FACTOR
- **NP1016_APB_CLK**: gated version of SYS_CLK that goes to the APB part of the NP1016 and that can be shut down when GPS is not needed.
- **UART1_CLK, UART2_CLK, TIMER_CLK, SPIM_CLK, SPIS_CLK, GPIO_CLK, DSU_CLK, CPU_CLK** and **MC_CLK** (memory controller) are gated clock version of SYS_CLK that individually clock the corresponding sub-blocks.

The gating of these clocks is controlled by a glitch free mechanism. The user has the ability to enable or disable each sub-clock individually by setting the corresponding CLK_EN bit in the CCTRL_REG of the system control block.

Moreover, each APB peripheral can dynamically require the activation of its own sub-clock. This mechanism achieves major power savings since the sub-clocks are activated only when needed by the respective blocks. For each subclock, the “clock demand” feature is enabled by a dedicated bit (DMD_EN) in the CCTRL_REG of the system control block.

When the two bits (CLK_EN and DMD_EN) of a given block are at '1', then the sub-block can generate a demand signal that activates its own sub-clock.

This mechanism is used by the UART controllers to finish the transmission even if the CPU is asked to deactivate the clock. The same mechanism applies to a UART that is starting to receive data, when no clock is present. The demand signal permits to get the UART clock running, to generate an interrupt and resume the CPU activity.

The NP1016 clocking requires peripherals to be used with a '11' (for CLK_EN and DMD_EN) configuration for both the NP1016_APB_CLK and the NP1016_GPS_CLK, in order for the low power features to be active. If the CLK_EN bit is cleared for these clocks, then the NP1016 has always all clocks inactive, independently from the number of active tracking modules.

Setting cpu_clock_demand enables the cpu clock to be halted, however the cpu clock will be maintained until the current memory access has been completed, therefore after setting this flag the cpu should execute nop’s to allow any pending accesses to complete. After this the cpu clock will be stopped until an interrupt occurs, this will clear the cpu_clock_demand flag enabling the cpu to continue execution, which will begin with the execution of the trap handler for the pending interrupt.

For the temperature sensor block an enable signal that is the pulse deleted version of factor 1024 of GPS_REF_CLK is generated. It is used by the temperature sensor interface to drive the slow analog blocks of the sensor (DAC + comparator). This signal is also used to implement the ageing mechanism in the UARTs and SPI slave FIFOs.

6.3 Sub-Clock Domains

Clock Domain	Demand signal generation
GPS_REF_CLK	Demand signal always at '1'[1]
NP1016_APB_CLK	Demand signal based on the read/write activity on APB
NP1016_GPS_CLK	Demand signal based on the transfer activity on the APB
UART1_CLK	Receive/transmit + edge detection
UART2_CLK	Receive/transmit + edge detection
TIMER_CLK	Demand signal for debug only. DMD_EN should not be set for the TIMER_CLK
SPIM_CLK	Demand signal based on the read/write activity on APB
SPIS_CLK	Demand signal always at '1'[1]
DSU_CLK	Demand signal controlled by DSUEN
MC_CLK	Not in idle state, DMD_EN bit cleared by an irq pending
CPU_CLK	MC not in idle state, DMD_EN cleared by an irq pending
GPIO_CLK	Demand signal always at '1'[1]

Note 1: Demand signal always at '1' means that the DMD_EN bit has no effect on the clock. As long as CLK_EN is set, the clock will be on.

7 System Control Block

7.1 System Control Registers

Address	Name	Type	Function
0x800000F0	SYS_CTRL_REG	Read/write	Multi purpose system control
0x800000F4	SYS_ID_REG	Read	NJ1030 version
0x800000F8	CCTRL_REG	Read/write	Sub-clocks enable/disable
0x800000FC	TEST_REG	Read/write	Test and debug control

The system control block gathers registers with various purposes.

The SYS_CTRL_REG controls the clock division factor, sets the system into SLEEP mode, selects the input to the ADC converter, the mode of the GPIO port and enables the watchdog counter overflow to reset the system. It entails also some flags indicating a low backup battery voltage, the validity of the PPS signal and the presence of an antenna at the RF front-end side.

The CCTRL register controls each internally generated sub clock domain with two bits: the DMD_EN (MSB) and CLK_EN bit (LSB). The CLK_EN has the highest priority and if set to '0' completely disable the sub clock domain. Once the sub clock domain has been activated (CLK_EN='1') the activity is controlled by the peripheral clocked by the sub clock domain. With DMD_EN = '0', the clock is always switched on, while if DMD_EN = '1', the clock is controlled by a synchronous signal generated by the peripheral. By default all clocks are switched on and demand mechanism is disabled ('01').

Clock control

DMD_EN	CLK_EN	Situation
X	0	Clock is always switched off
0	1	Clock is always switched on (default)
1	1	Clock is switched on by the peripheral served by the clock domain

For test purposes, two bits in the TEST_REG are available in order to speed up the RTC behavior. TEST_RTC_CLK permits to feed a SYS_CLK divided by 4 into the 15 bits prescaler. The TEST_RTC_PRESCALER put the prescaler into a divide by 8 mode, in order to accelerate the execution of a functional RTC test. (See RTC chapter)

7.2 Register Details

SYS_CTRL_REG (Addr 0x800000F0)

Field	Bits	Rst	Description
RESERVED	31:15	undef	Reserved bits
SYS_SPI_SLAVE	14	0	SPI master slave select. Selects the mode of the SPI external signals. Default is 0, Master.
SYS_ANT_OK	13	undef	Antenna OK. The value delivered by the RF front end is sampled into this register after three SYS_CLK cycles. '1' means that the antenna is present and working properly.
SYS_WD_RST_EN	12	0	Watchdog reset enable. If set, enables the execution of a system reset, when the watchdog counter overflows.
SYS_GPIO_MODE	11:9	000	Selects the mode of the GPIO port.
SYS_ADC_SEL	8:6	000	Select the input to the ADC converter. Default (000) is the on chip temperature sensor, while '001' to '111' select AIN[0] to AIN[6].
SYS_PPS_VALID	5	0	Pulse Per Second Valid flag. Set to '1' by the GPS SW stack when the NP1016 PPS signal is generated and correctly aligned to the GPS time.
SYS_LOW_BAT	4	0	Low Backup Battery Flag. It is set when the VBAT signal is lower than 1.10V.
SYS_GO_SLEEP	3	0	Go to Sleep. When set, this bit forces the system into sleep mode.
SYS_CLK_DIV	2:0	111	Clock division factor for the generation of the GPS_REF_CLK / NP1016_GPS_CLK. Default is 111 = div by 8, 110 = div by 7 until 000 = div by 1.

SYS_ID_REG (Addr 0x800000F4)

Field	Bits	Rst	Description
SYS_ID	32	0x08012010	NJ1030 version.

CCTRL_REG (Addr 0x800000F8)

NB: For each pair of bit of this register, the MSB is the DMD_EN bit and the LSB is the CLK_EN bit, as explained in the 'Clock control' table.

Field	Bits	Rst	Description
RESERVED	31:24	undef	Reserved bits
CCTRL_GPS_REF	23:22	01	External GPS reference clock control
CCTRL_NP1016_APB_CLK	21:20	01	Control of the clock of the APB interface of the NP1016
CCTRL_NP1016_GPS_CLK	19:18	01	Control of the GPS clock of the NP1016
CCTRL_UART2	17:16	01	UART2 clock control
CCTRL_UART1	15:14	01	UART1 clock control
CCTRL_TIMER	13:12	01	TIMERs clock control. DMD_EN should never be set. Debug only.
CCTRL_SPIM	11:10	01	SPI mater clock control
CCTRL_SPIS	9:8	01	SPI slave clock control
CCTRL_DSU	7:6	01	DSU clock control
CCTRL_MC	5:4	01	Memory controller clock control
CCTRL_CPU	3:2	01	CPU clock control
CCTRL_GPIO	1:0	01	GPIO clock control

TEST_REG (Addr 0x800000FC)

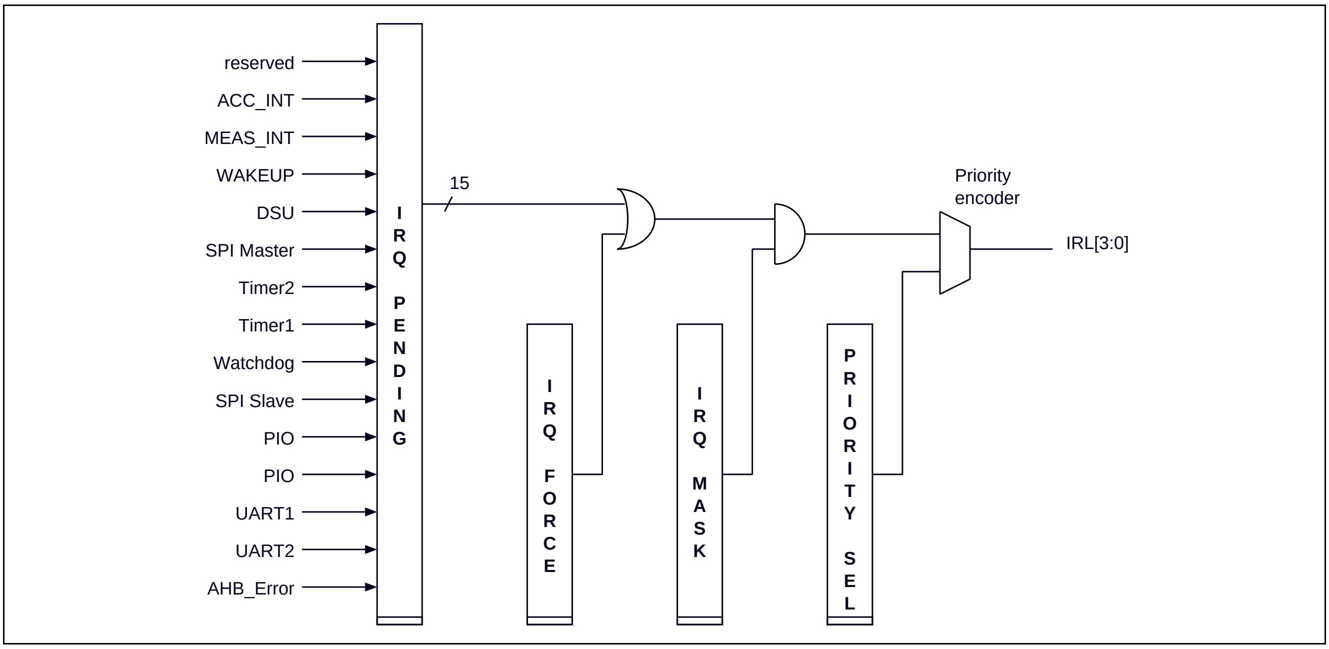
Field	Bits	Rst	Description
RESERVED	31:7	undef	Reserved bits
TEST_SPI	6	0	SPI test mode. When set, data from the slave is routed back to the master so that a back to back test of the master and slave devices can be performed.
TEST_DMD_SEL	5:2	0000	Selects the internal clock demand signal to be muxed out to the DMD_OUT signal on the GPIO.
TEST_RTC_CLK	1	0	If set, SYS_CLK divided by 8 is fed to the input of the RTC prescaler
TEST_RTC_PRESCALER	0	0	If set, the RTC prescaler is set into a divide by 8 mode.

8 Interrupt Controller

8.1 Interrupt Control Registers

Address	Name	Type	Function
0x80000090	IRQMASK_REG	Write	Interrupt mask and priority register
0x80000094	IRQPEND_REG	Read	Interrupt pending register
0x80000098	IRQFORCE_REG	Read/Write	Interrupt force register
0x8000009C	IRQCLEAR_REG	Write	Interrupt clear register

Figure 6. Interrupt controller block diagram



Interrupt assignments

IRQ number	Source
15	Reserved
14	ACC_INT from NP1016
13	MEAS_INT from NP1016
12	CPU_WAKE (RTC_WAKE or EXT_WAKE)
11	DSU
10	SPI Master
9	Timer 2
8	Timer 1
7	Watchdog
6	SPI Slave
5	PIO interrupt – programmable
4	PIO interrupt – programmable
3	UART1
2	UART2
1	AHB error

The LEON interrupt controller is used to prioritise and propagate interrupt requests from internal or external devices to the integer unit. In total 15 interrupts are handled, divided on two priority levels.

8.2 Operation

When an interrupt is generated, the corresponding bit is set in the interrupt pending register. The pending bits are ANDed with the interrupt mask register and then forwarded to the priority selector. Each interrupt can be assigned to one of two priority levels as programmed in the interrupt level register. Level 1 has higher priority than level 0. The interrupts are prioritised within each level, with interrupt 15 having the highest priority and interrupt 1 the lowest. The highest interrupt from level 1 will be forwarded to the Integer Unit - if no unmasked pending interrupt exists on level 1, then the highest unmasked interrupt from level 0 will be forwarded. When the integer unit acknowledges the interrupt, the corresponding pending bit will automatically be cleared.

Interrupt can also be forced by setting a bit in the interrupt force register. In this case, the IU acknowledgement will clear the force bit rather than the pending bit.

8.3 Register Details

IRQMASK_REG (Addr 0x80000090)

Field	Bits	Rst	Description
ILEVEL	31:17	0x0	Interrupt level - indicates whether an interrupt belongs to priority level 1 (ILEVEL[n]=1) or level 0 (ILEVEL[n]=0). Bit 31 corresponds to IRQ15, Bit 30 to IRQ14, and so on.
RESERVED	16	undef	Reserved bit
IMASK	15:1	0x0	Interrupt mask - indicates whether an interrupt is masked (IMASK[n]=0) or enabled (IMASK[n]=1).
RESERVED	0	undef	Reserved bit

IRQPEND_REG (Addr 0x80000094)

Field	Bits	Rst	Description
RESERVED	31:16	undef	Reserved bits
IPEND	15:1	0x0	Interrupt pending (IPEND[15:1]) - indicates whether an interrupt is pending (IPEND[n]=1).
RESERVED	0	undef	Reserved bits

IRQFORCE_REG (Addr 0x80000098)

Field	Bits	Rst	Description
RESERVED	31:16	undef	Reserved bits
IFORCE	15:1	undef	Interrupt force (IFORCE[15:1]) - indicates whether an interrupt is being forced (IFORCE[n]=1).
RESERVED	0	undef	Reserved bits

IRQCLEAR_REG (Addr 0x8000009C)

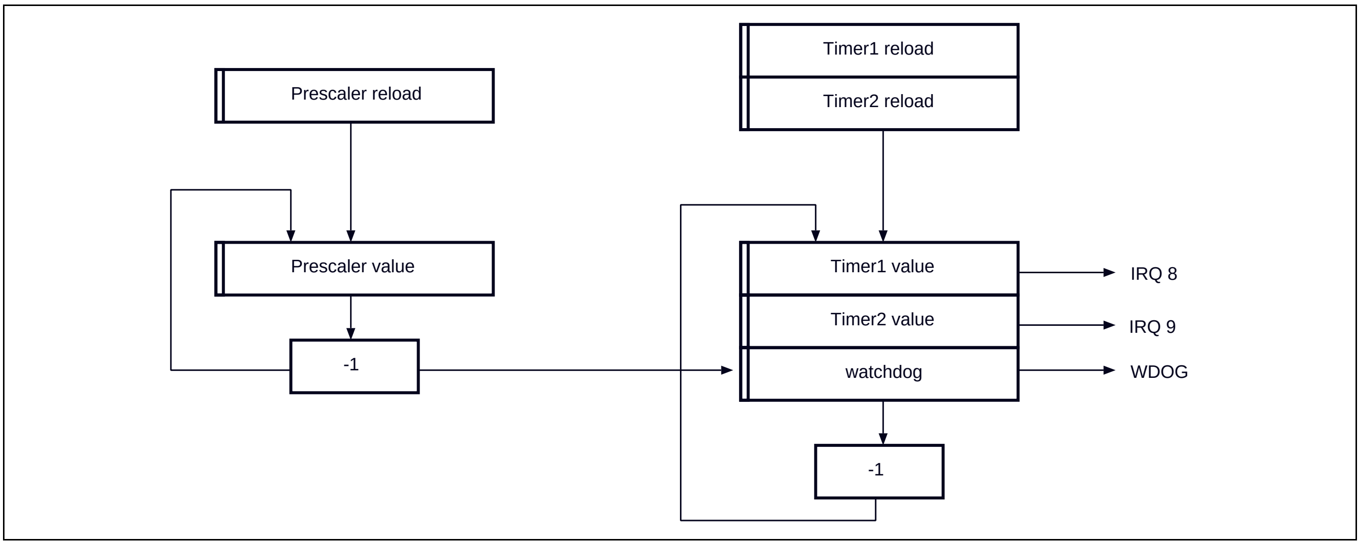
Field	Bits	Rst	Description
RESERVED	31:16	undef	Reserved bits
ICLEAR	15:1	0x0	Interrupt clear (ICLEAR[15:1]) - if written with a ‘1’, ICLEAR[n] will clear IPEND[n] in the interrupt pending register. A consecutive read on ICLEAR[n] will return ‘0’ and confirm that the pending has been cleared.
RESERVED	0	undef	Reserved bits

9 Timer and Watchdog

9.1 Timer Control Registers

Address	Name	Type	Function
0x80000040 / 0x80000050	TIMER_CNTx_REG	Read	Timer value
0x80000044 / 0x80000054	TIMER_LOADx_REG	ReadWrite	Timer reload value
0x80000048 / 0x80000058	TIMER_CTRLx_REG	Read/Write	Timer general control register
0x8000004C	WDOG_CNT	Read	Watchdog value
0x80000060	PRESCALER_CNT_REG	Read	Prescaler counter
0x80000064	PRESCALER_LOAD_REG	Read/write	Prescaler reload

Figure 7. Timer unit block diagram



The timer unit implements two 24-bit timers, one 24-bit watchdog and one 10-bit shared prescaler.

9.2 Operation

The prescaler is clocked by the system clock and decremented on each clock cycle. When the prescaler underflows, it is reloaded from the prescaler reload register and a timer tick is generated for the two timers and watchdog. The effective division rate is therefore equal to prescaler reload register value + 1.

The operation of the timers is controlled through the timer control register. A timer is enabled by setting the enable bit in the control register. The timer value is then decremented each time the prescaler generates a timer tick. When a timer underflows, it will automatically be reloaded with the value of the timer reload register if the reload bit is set, otherwise it will stop (at 0x000000) and reset the enable bit. An interrupt will be generated after each underflow.

The timer can be reloaded with the value in the reload register at any time by writing a ‘1’ to the load bit in the control register.

The watchdog operates similar to the timers, with the difference that it is always enabled. The watchdog output is connected to the reset control block.

For SW debugging the generation of the reset by the watchdog can be disabled by clearing the SYS_WD_RST_EN bit in the SYS_CTRL_REG which allows to enable/disable the watchdog and trap the event leading to the watchdog overflow, avoiding a processor reset.

To minimise complexity, the two timers and watchdog share the same decrementer. This means that the minimum allowed prescaler division factor is 4 (reload register = 3).

9.3 Register Details

TIMER_CNTx_REG (Addr 0x80000040 / 0x80000050)

Field	Bits	Rst	Description
RESERVED	31:24	undef	Reserved bits
TIMERx_VALUE	23:0	undef	Timer value

TIMER_LOADx_REG (Addr 0x80000044 / 0x80000054)

Field	Bits	Rst	Description
RESERVED	31:24	undef	Reserved bits
TIMERx_LOAD	23:0	undef	Timer reload value

TIMER_CTRLx_REG (Addr 0x80000048 / 0x80000058)

Field	Bits	Rst	Description
RESERVED	31:3	undef	Reserved bits
TIMERx_LD	2	undef	Load counter. When set, will load the timer reload register into the timer counter register. Always read as a ‘0’.
TIMERx_RL	1	undef	Reload counter. If set, then the counter will automatically be reloaded with the reload value after each underflow.
TIMERx_EN	0	undef	Enable. Enables the timer when set.

WDOG_CNT_REG (Addr 0x8000004C)

Field	Bits	Rst	Description
RESERVED	31:24	undef	Reserved bits
WDOG_VALUE	23:0	undef	Watchdog counter value

PRESCALER_CNT_REG (Addr 0x80000060)

Field	Bits	Rst	Description
RESERVED	31:10	undef	Reserved bits
PRESC_CNT	9:0	undef	Prescaler counter value

PRESCALER_LOAD_REG (Addr 0x80000064)

Field	Bits	Rst	Description
RESERVED	31:10	undef	Reserved bits
PRESC_LOAD	9:0	undef	Prescaler reload value

10 Real Time Clock (RTC)

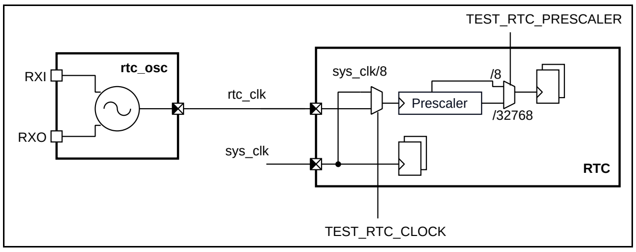
10.1 RTC Control Registers

Address	Name	Type	Function
0x80008000	RTC_TIME_REG	Read/Write	Real Time Clock
0x80008004	RTC_WAKEUP_REG	Write	Wake up value

Clocked by a dedicated 32.768 kHz oscillator the RTC is battery backed up, together with the NVRAM. The RTC has 1s resolution and doesn't overflow when the maximal value of 2^{30} seconds is reached. The RTC is built with a 15 bits ripple carry prescaler that divides the 32.768 kHz clock followed by a synchronous 30 bit programmable counter.

For test purposes the rtc_clock can be sped-up using the TEST_RTC_CLK and the TEST_RTC_PRESCALER bits of the TEST_REG (see ‘System Control block’ section)

Figure 8. RTC clock generation



The RTC has a precision of 1 second. The alarm function can be programmed by writing the wake-up time into a 30 bit register (RTC_WAKEUP_REG). If the RTC value is the same or bigger than the alarm value, then the RTC_WKUP event is generated, which is usually used to recover from a sleep mode. The RTC_WKUP is also connected to the CPU interrupt controller.

The CPU can set the RTC time by writing to the RTC_TIME_REG register and gets the RTC value by reading from the same address. The RTC_TIME_REG is automatically updated with the latest RTC counter value (RTC_VALUE) at each tick of the RTC clock. This enables the RTC to be instantly read by the CPU at any time, as it is never more than 1 second out.

The RTC_TIME_REG register is double buffered so that it can be written at system clock speed. However the value is transferred from the system accessible RTC_TIME_REG register to the RTC internal register synchronously to the 32 KHz RTC Clock.

As physically writing to the RTC counter may take up to 1.5 RTC clock cycles, the APB_RTC logic is designed to handle this latency automatically. The RTC_TIME_REG register can be updated without taking care of the timing of the RTC counter. The RTC_WRFLAG acts as a write completed flag. If RTC_WRFLAG is set by the software, it will be automatically cleared when the update of the RTC counter with the new RTC_VALUE has been completed. Then, the RTC_WRFLAG can be polled in the interval to check the progress of the write.

A wake up value can be set with the RTC_WAKEUP_REG register. A value of the RTC counter equal or higher than the RTC_WAKEUP_TIME value will set the RTC_WKUP bit in the RTC_TIME_REG register and send a wake-up signal to the powerup unit. RTC_WKUP is also connected to IRQ 12 of the interrupt controller.

Accesses to the RTC_WAKEUP register occur at system clock speed, this register is not shadowed and comparisons to the RTC_TIME_REG are performed in the RTC clock domain, hence it is possible for updates of the RTC_WAKEUP_REG register to cause a spurious RTC_WKUP event. Therefore the RTC wakeup interrupt must be masked when updating the RTC_WAKEUP_REG register.

10.2 NVRAM

The 2k x 32 bit battery backed-up SRAM is mapped at addresses 0x80002000 - 0x80003FFF in the APB space and is only accessible through 32 bits data words.

Data integrity as well as safe power supply switching to VBAT on failing DVDD or LVDD power supply is guaranteed by the Power Supervisor block.

The internal POWER_OK signal generated by the Power Supervisor block is used to disable any access from the CPU during the transition phase from valid power supply to invalid power supply.

However if the switching to VBAT occurs during a write operation to the NVRAM, invalid or partially written data blocks could be stored in the NVRAM. Hence, the SW routines for the NVRAM access must implement a data integrity check mechanism that detects corrupted data blocks.

10.3 Power Fence

A power fence logic is introduced between the APB interface and the part consisting of the NVRAM macro and the RTC circuits. The power fence guarantees that with any voltage between 0V and LVDD on one side and any voltage between 0V and LVDD max or 0 and VBAT max on the other, no gates are damaged. When power is lost or an asynchronous reset occurs, the power fence forces to a pre-determined level the internal NVRAM control signals while all outputs from the memory and RTC are gated to 0.

10.4 Register Details

RTC_TIME_REG (Addr 0x80008000)

Field	Bits	Rst	Description
RTC_WRFLAG	31	0	“Write has occured” flag. This flag is automatically cleared when the RTC_VALUE has been written to the RTC counter. It can be polled to check if the write operation is completed.
RTC_WKUP	30	0	Wake-up match flag. Indicates that the RTC wake-up value has been reached (i.e RTC_VALUE ≥ RTC_WAKEUP_TIME). Must be cleared manually.
RTC_VALUE	29:0	NA	RTC counter value. The value written here will be transferred to the RTC counter at the next edge of the RTC clock. These bits also reflect the current value of the RTC counter.

RTC_WAKEUP_REG (Addr 0x80008004)

Field	Bits	Rst	Description
RESERVED	31:30	undef	Reserved bits
RTC_WAKEUP_TIME	29:0	0x3FF FFFFF	Wakeup match value. It is continually compared with the value of the RTC counter. If RTC_WAKEUP_TIME ≥ RTC counter, then a RTC wake-up event is generated.

11 GPS Correlation Processor

11.1 GPS Correlator Global Control Registers

Address	Name	Size	Type	Function
0x80001800	TIMING_DUTY_CYCLE_REG	14	Write	Sets the ACC_INT period and the duration of the LPM1 ON phase
0x80001804	DOZE_PERIOD_REG	16	Write	Defines the duration of the OFF phase of the LPM2 mode
0x80001808	FE_WAKE_UP_REG	16	Write	Defines the RF front end wake up time in the LPM1 and LPM2 modes
0x8000180C	PPS_DIVIDE_REG	6	Write	Division factor applied to MEAS_INT for the generation of a PPS signal
0x80001810	MEAS_PERIOD_LOW_REG	16	Write	Low portion of the MEAS_INT duration definition
0x80001814	MEAS_PERIOD_HIGH_REG	9	Write	High portion of the MEAS_INT duration definition
0x80001818	BB_CTRL_REG	9	Write	Control Register with different functions
0x8000181C	ACTIVE_TMS_REG	16	Write	Individually activated the tracking modules
0x80001820	INT_FLAG_CLR_REG	2	Write	Clears the ACC_INT and MEAS_INT flags
0x80001824	TM_CLEAR_REG	16	Write	Individually clears the status of the tracking modules
0x8000183C	BB_TEST_REG	1	Write	Sets the global registers in test mode
0x80001840	STATUS_REG_A	16	Read	Indicates new dumped values in the individual tracking modules
0x80001844	STATUS_REG_B	2	Read	Indicate that ACC_INT and MEAS_INT occurred
0x80001848	STATUS_REG_C	16	Read	Indicate that a dump occurred in the individual tracking modules, after a MEAS_INT
0x8000187C	VERSION	16	Read	NP1016 Version Number

The GPS correlation processor is an instantiation of the NP1016 correlation processor with 16 tracking modules (TMs). It processes the digital IF signal delivered by the RF front end, correlating it with replicas of the expected GPS signal and producing the measurements data that are used to compute the GPS receivers time and position. Moreover the unit generates two control signals (FE_P0_INT and FE_P1_INT) used to control the power management modes of the RF front end. Optionally a pulse per second (PPS) signal is generated once the GPS time is known by the system. The correlation unit generates two interrupt signals (ACC_INT and MEAS_INT) that are available on the interrupt controller.

11.2 Processor Interface

It has a set of registers for each of the 16 tracking modules and a set of global registers. Each tracking module has its own set of registers that are mapped according to the following table.

TM register mapping

TM Number	Address Space
0	0x80001000
1	0x80001080
2	0x80001100
3	0x80001180
4	0x80001200
5	0x80001280
6	0x80001300
7	0x80001380
8	0x80001400
9	0x80001480
10	0x80001500
11	0x80001580
12	0x80001600
13	0x80001680
14	0x80001700
15	0x80001780

The following table describes which registers are available for each tracking module, within the individual address range.

11.3 Individual Tracking Module registers

Name	Address	Size	Type	Description
CARRIER_NCO_INCR_LOW	0x00	16	Write	Low portion of the 30 bit carrier NCO increment
CARRIER_NCO_INCR_HIGH	0x04	14	Write	High portion of the 30 bit carrier NCO increment
CODE_NCO_INCR_LOW	0x08	16	Write	Low portion of the 30 bit code NCO increment
CODE_NCO_INCR_HIGH	0x0C	14	Write	High portion of the 30 bit code NCO increment
SLEW_CTRL	0x10	12	Write	Controls the slewing of the C/A code generation
TM_MODE	0x14	16	Write	TM mode configuration
EPOCHS_SET	0x18	12	Write	Sets the GPS epochs counters
INTEGR_Q_PROMPT	0x40	16	Read	Integrated value on the Q-prompt path
INTEGR_Q_EL	0x44	16	Read	Integrated value on the Q-early/late path
INTEGR_I_PROMPT	0x48	16	Read	Integrated value on the I-prompt path
INTEGR_I_EL	0x4C	16	Read	Integrated value on the I-early/late path
CARRIER_CYCLE_CNT_LOW	0x50	16	Read	Low portion of the carrier cycles counter
CARRIER_CYCLE_CNT_HIGH	0x54	14	Read	High portion of the carrier cycles counter
CARRIER_NCO_PHASE	0x58	16	Read	Carrier NCO phase
CODE_NCO_PHASE	0x5C	16	Read	Code NCO phase
CODE_PHASE	0x60	10	Read	C/A code phase
EPOCHS	0x64	12	Read	GPS epochs status

11.4 IF Interface

The signal path input is the two bit IF signal generated by the GPS RF front-end (SGN and MAG) with the coding described in the following table. If the RF front-end produces a one bit signal, then only the SGN bit is used, while the MAG input is tied-up.

IF signal conversion

IF value	SGN	MAG
3	0	1
1	0	0
-1	1	0
-3	1	1

11.5 Clocking

The GPS correlation engine is clocked with two clocks: the NP1016_APB_CLK applied to the interface to the processor and NP1016_GPS_CLK that is a pulse deleted version of the SYS_CLK and is used for the GPS processing. The pulse deletion factor is the same as the clock division factor used to generate the external GPS reference clock for the RF front end. In most of the portion clocked by SYS_CLK, the clock on demand option is available. If activated, this clocking option generates a clock pulse only when an access to the interface registers is executed by the CPU.

11.6 Functional Behaviour

Starting from the IF signal, each Traking Module can be assigned to acquire or track the signal from a selected space vehicle.

Each tracking module entails:

- a carrier NCO (numerically controlled oscillator) generating I and Q components.
- a code NCO feeding a C/A code generator that produces the C/A code corresponding to a given space vehicle and a version of the same code spaced by ½ code symbol (chip). The C/A code generator also offers the option to slew the C/A code generation by a programmable number of symbols.

The incoming IF signal is multiplied by the I and Q carrier components and by the two version of the C/A code. The resulting 4 signal pathes are continuously accumulated and the accumulated value is periodically dumped into 4 registers (INTEGR_Q_P, INTEGR_Q_EL, INTEGR_I_EL and INTEGR_I_P). These 4 values indicate the level of correlation between the locally generated signal and the one of the space vehicle that the TM has been assigned to.

Beside the correlation values, each TM generates a set of 5 measurements consisting of the code NCO phase, the carrier NCO phase, the carrier cycle count, the C/A code phase and the number of GPS epochs (1ms and 20ms epochs).

The GPS SW stack activates the necessary number of TMs, assigns them to space vehicles and does a C/A code and frequency scan in order to acquire the space vehicles signals. Once the space vehicle is acquired, the correlation values are monitored and the carrier and code NCO generators are continuously adjusted in order to track the acquired signal. By extracting GPS data from the tracked signals and computing pseudo ranges starting from the 5 measurements, the GPS SW can then compute position, velocity and time (PVT) of the receiver.

In order to operate the GPS SW has to continuously serve the ACC_INT interrupt, while it can set the MEAS_INT timing to a lower rate and poll the occurrence of this signal.

11.7 RF Power Management Signals

The GPS correlator unit generates two internal signals (FE_P1_INT and FE_P0_INT) that are used to control the power management features of the NJ100x RF front ends. The power modes are coded as follow:

NJ100x power modes

NJ100x Mode	FE_P1_INT	FE_P0_INT
Fully Active	1	0
Stand-by	1	1
Doze	0	1
Off	0	0

Before being made available on the external FE_P1 and FE_P0 pins, these signals are combined with the NJ1030 operating mode as described in the following paragraph.

11.8 Control of the FE_P0 and FE_P1 Signals

The logic in the reset block has to control the FE_P0 and FE_P1 signals. The FE_P0 and FE_P1 output pins are asynchronously overridden for certain system and reset events.

These are listed here in order of decreasing priority, i.e the first in the table is the highest priority. fe_p0_in and fe_p1_in are the signal generated by the internal NP1016 block which in some cases are overridden, depending on the system status and origin of the clock. A detailed description of the NP1016 and its corresponding registers is available in the NP1016 datasheet.

FE_P0 and FE_P1 control

EXT_RESETN	CLK_SOURCE	fe_p1_in	fe_p0_in	FE_P1	FE_P0	Status
X	X	X	X	0	0	Sleep mode
X	X	X	X	0	0	Power fail (LVDD=0)
0	X	X	X	1	0	During reset
1	0	0	0	1	0	Prevent accidental turn off of external osc.
1	0	0	1	0	1	Pass through NP1016
1	0	1	0	1	0	Pass through NP1016
1	0	1	1	1	1	Pass through NP1016
1	1	0	0	0	0	Pass through NP1016
1	1	0	1	0	1	Pass through NP1016
1	1	1	0	1	0	Pass through NP1016
1	1	1	1	1	1	Pass through NP1016

12 External Memory Support

12.1 EBI Control Registers

Address	Name	Type	Function
0x80000000	MEM_MCFG1_REG	Read/Write	RAM4 and I/o control
0x80000004	MEM_MCFG2_REG	Read/write	Wait states and memory access
0x80000008	MEM_MCFG3_REG	Read/write	Data width

The external bus interface (EBI) supports 4 banks of asynchronous memory with individually programmable number of wait states: each bank is 16 MBytes wide (24, 23 or 22 bits address depending on data width) and can be cached.

Programmable data width of 8, 16 and 32 bits as well as read-modify-write operation is supported. A fifth bank (RAM3) can be accessed through the GPIO interface. A sixth bank for asynchronous I/O (optionally cachable) is also available. The EBI interface supports state-of-the-art SRAM and Flash memory devices.

- Individual wait state programming (0 to 7 for memories and 0 to 15 for I/O space)
- Access to 8/16/32 bit word as well as with read-modify-write operation.

In addition, RAM4 has an extra write protection bit (MEM_RAM4_WEN), with respect to the other 3 memory spaces.

An I/O channel provides access to a general purpose I/O space. Two different addressing spaces are possible for I/O, depending on the need to cache the access or not.

12.2 External Memory Access

Adress space	Size	Mapping	Chip select	Cacheable
0x30000000 - 0x6FFFFFFF	16M	IO(echo)	IOSN	Instruction only
0x21000000 - 0x2FFFFFFF	16M	IO(echo)	IOSN	NO
0x20000000 - 0x20FFFFFF	16M	IO	IOSN	NO
0x10000000 - 0x10FFFFFF	16M	RAM4	CSN4	Instruction and Data
0x03000000 - 0x03FFFFFF	16M	RAM3	CSN3[1]	Instruction and Data
0x02000000 - 0x02FFFFFF	16M	RAM2	CSN2	Instruction and Data
0x01000000 - 0x01FFFFFF	16M	RAM1	CSN1	Instruction and Data
0x00000000 - 0x00FFFFFF	16M	RAM0	CSN0	Instruction and Data

Note 1: Available through GPIO

12.3 Address Multiplexing of External Memories

Address multiplexing is provided internally to facilitate jumperless selection of 8/16/32 bit external memory devices using a single PCB. This allows a single PCB to be populated according to the desired cost / performance model. The address multiplexing will route the address from the core CPU to the address pins of the device such that A0 of the NJ1030 can always be connected to A0 of the external component. These address lines are multiplexed dynamically according to the width specified in the control register for the memory area being accessed.

Memory width	EBI.Addr[23]	EBI.Addr[22]	EBI.Addr[21..0]
8	CPU.Addr[23]	CPU.Addr[22]	CPU.Addr[21..]
16	Not connected	CPU.Addr[23]	CPU.Addr[22..1]
32	Not connected	Not connected	CPU.Addr[23..2]

12.4 Data Multiplexing of External Memories

Data is supplied to the CPU on dedicated data lines, as described in the following table. Byte 0 is the least significant byte in a 32 bits word; byte 3 is the most significant.

Write lines are associated with the external memories as described in the following table. After Reset the system defaults to treating the external memory as 8 bits and discarding any data on EBI.Data[23..0]. A different NJ1030 startup code for 8,16, and 32 bit memory systems correctly configures the memory controller according to the width of the device fitted, thus the contents of the external flash memory can be used to configure the memory width without using any jumpers on the PCB. A bus keeper function is provided so that unused data lines may be left floating without damage.

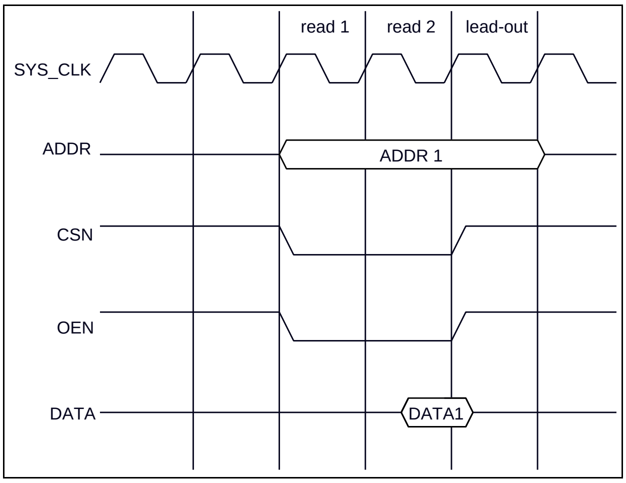
Memory width	EBI.Data[31..24]	EBI.Data[23..16]	EBI.Data[15..8]	EBI.Data[7..0]
8	RWEN[0]	-	-	-
16	RWEN[0]	RWEN[1]	-	-
32	RWEN[0]	RWEN[1]	RWEN[2]	RWEN[3]

12.5 SRAM Access

12.5.1 Typical Read Access

A read access to SRAM consists of two data cycles and between zero and seven waitstates. On non-consecutive accesses, a lead-out cycle is added after a read cycle to prevent bus contention due to slow turn-off time of memories or I/O devices. Figure 9 shows the basic read cycle waveform (zero waitstate).

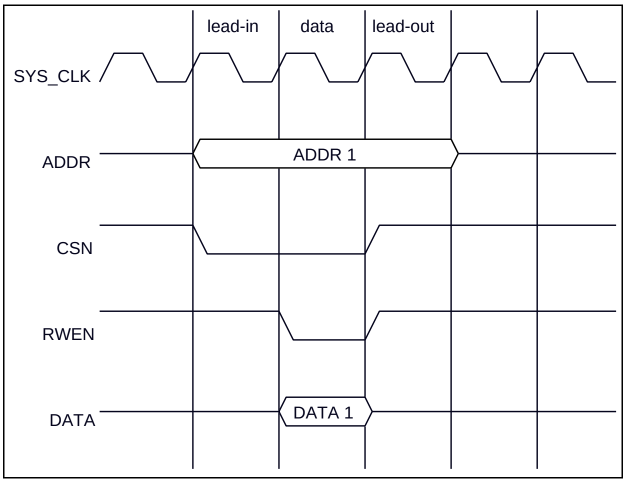
Figure 9. Typical read access



12.5.2 Typical Write Access

A write access is similar to the read access but takes a minimum of three cycles. Through a feedback loop from the write strobes, the data bus is guaranteed to be driven until the write strobes are de-asserted. Each byte lane has an individual write strobe to allow efficient byte and half-word writes. If the memory uses a common write strobe for the full 16- or 32-bit data, the read-modify-write bit MCR2 should be set to enable read-modify-write cycles for sub-word writes.

Figure 10. Typical write access



12.5.3 Burst Cycles

To improve the bandwidth of the memory bus, accesses to consecutive addresses can be performed in burst mode. Burst transfers will be generated when the memory controller is accessed using an AHB burst request. These include instruction cache-line fills, double loads and double stores. The timing of a burst cycle is identical to the programmed basic cycle with the exception that during read cycles the lead-out cycle will only occur after the last transfer.

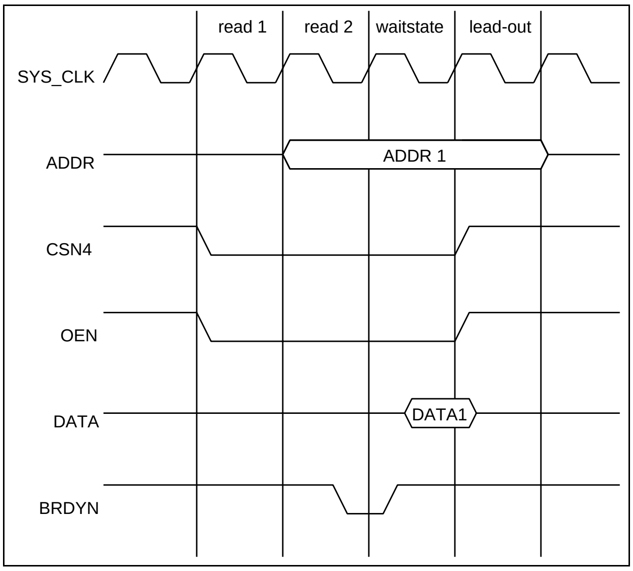
12.5.4 Using BRDYN

BRDYN input signal can be used to stretch access cycles to the I/O area and the ram area decoded by CSN4. The accesses using BRDYN have the following sequence:

- Wait for BRDYN to be driven low by the peripheral.
- Wait for the pre-programmed number of wait states as defined in the control registers and perform the read.

BRDYN functionality can be enabled separately for RAM (MEM_MCFG2_REG) or I/O (MEM_MCFG1_REG) areas.

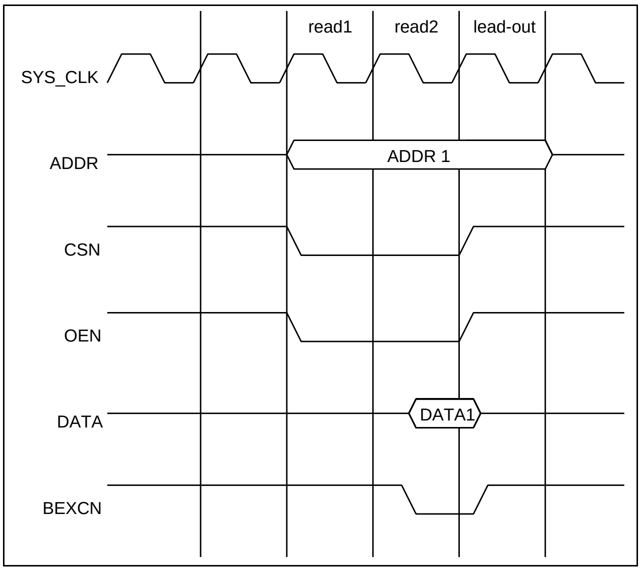
Figure 11. RAM read - 1 BRDYN controlled wait state.



12.5.5 Access Errors

An access error can be signalled by asserting the BEXCN input signal, which is sampled together with the data. If the usage of BEXCN is enabled in MEM_MCFG1_REG, an error response will be generated on the internal AMBA bus. BEXCN can be enabled or disabled through MEM_MCFG1_REG, and is active for all areas (I/O and RAM).

Figure 12. Read cycle with BEXCN.



12.6 Register Details

MEM_MCFG1_REG (Addr 0x80000000)

Field	Bits	Rst	Description
RESERVED	31:29	undef	Reserved bits.
MEM_IO_WIDTH	28:27	10	I/O space data width. Defines the data width of the I/O area ('00'=8 bits, '01'=16 bits, '10'=32 bits)
MEM_IO_RDY	26	0	I/O bus ready enable. If set, will enable the BRDYN pin (available through GPIO). BRDYN can be used for interfacing slow peripherals for which the max number of wait states isn't sufficient
MEM_IO_ERR	25	0	I/O bus error (BEXCN available through GPIO) enable.
RESERVED	24	undef	Reserved bit.
MEM_IO_WS	23:20	1111	I/O wait states. Default is '1111', 15 wait states.
MEM_IO_EN	19	0	I/O enable. If set, the access to the I/O bus area is enabled.
RESERVED	18:12	undef	Reserved bits.
MEM_RAM4_WEN	11	0	Write protection bit to the RAM4 space. If sets, enables write cycles in RAM4.
RESERVED	10	undef	Reserved bit.
MEM_RAM4_WIDTH	9:8	00	RAM4 data width. ('00'=8 bits, '01'=16 bits, '10'=32 bits)
MEM_RAM4_WWS	7:4	1111	RAM4 Write wait states. Default is '1111', 15 wait states.
MEM_RAM4_RWS	3:0	1111	RAM4 Read wait states. Default is '1111', 15 wait states.

MEM_MCFG2_REG (Addr 0x80000004)

Field	Bits	Rst	Description
MEM_RAM3_WWS	31:29	111	RAM3 Write wait states. Default is ‘111’, 7 wait states.
MEM_RAM3_RWS	28:26	111	RAM3 Read wait states. Default is ‘111’, 7 wait states.
MEM_RAM2_WWS	25:23	111	RAM2 Write wait states. Default is ‘111’, 7 wait states.
MEM_RAM2_RWS	22:20	111	RAM2 Read wait states. Default is ‘111’, 7 wait states.
MEM_RAM1_WWS	19:17	111	RAM1 Write wait states. Default is ‘111’, 7 wait states.
MEM_RAM1_RWS	16:14	111	RAM1 Read wait states. Default is ‘111’, 7 wait states.
MEM_RAM0_WWS	13:11	111	RAM0 Write wait states. Default is ‘111’, 7 wait states.
MEM_RAM0_RWS	10:8	111	RAM0 Read wait states. Default is ‘111’, 7 wait states.
RESERVED	7:4	undef	Reserved bits.
MEM_RAM_RDY	3	0	Memory ready enable. If set, will enable the BRDYN pin (available through GPIO). BRDYN can be used for interfacing slow peripherals for which the max number of wait states isn’t sufficient.
MEM_RMW	2	0	RAM read modify write. Enable read-modify-write cycles on sub-word writes to 16- and 32-bit areas with common write strobe (no byte write strobe).
RESERVED	1:0	undef	Reserved bits.

MEM_MCFG3_REG (Addr 0x80000008)

Field	Bits	Rst	Description
RESERVED	31:9	undef	Reserved bits.
MEM_MUX_OVERRIDE	8	0	Override bit for the address shifting mechanism. If set, no shifting of the data and address is carried out for different memory widths.
MEM_RAM3_WIDTH	7:6	10	RAM3 data width (‘00’ = 8bits, ‘01’ = 16bits, ‘10’ = 32bits)
MEM_RAM2_WIDTH	5:4	10	RAM2 data width (‘00’ = 8bits, ‘01’ = 16bits, ‘10’ = 32bits)
MEM_RAM1_WIDTH	3:2	10	RAM1 data width (‘00’ = 8bits, ‘01’ = 16bits, ‘10’ = 32bits)
MEM_RAM0_WIDTH	1:0	00	RAM0 data width (‘00’ = 8bits, ‘01’ = 16bits, ‘10’ = 32bits)

13 UARTs

13.1 UARTs Control Registers

Address	Name	Type	Function
0x80000070 / 0x80000080	UARTx_DATA_REG	Read/Write	Data register
0x80000074 / 0x80000084	UARTx_STAT_REG	Read	Status register
0x80000078 / 0x80000088	UARTx_CTRL_REG	Write	Control register
0x8000007C / 0x8000008C	UARTx_USC_REG	Write	Scaler reload values

Transmit

The transmitter is enabled through the TX_EN bit in the UART control register. When ready to transmit, data is transferred from the transmitter holding register to the transmitter shift register and converted to a serial stream on the transmitter serial output pin (TX). The UART automatically sends a start bit followed by eight data bits, an optional parity bit, and one stop bit. The least significant bit of the data is sent first.

Receive

The receiver is enabled for data reception through the receiver enable (RX_EN) bit in the UART control register. The receiver looks for a high to low transition of a start bit on the receiver serial data input pin. If a transition is detected, the state of the serial input is sampled a half bit clock later. If the serial input is sampled high the start bit is invalid and the search for a valid start bit continues. If the serial input is still low, a valid start bit is assumed and the receiver continues to sample the serial input at one bit time intervals (at the theoretical centre of the bit) until the proper number of data bits and the parity bit have been assembled and one stop bit has been detected. The serial input is shifted through an 8-bit shift register where all sampled signals have to have the same value before the new value is taken into account, effectively forming a low-pass filter with a cut-off frequency of 1/8 system clock.

During reception, the least significant bit is received first. The data is then transferred to the receiver holding register and the data ready (READY) bit is set in the USART status register. The parity and framing error bits are set at the received byte boundary at the same time as the receiver ready bit is set.

Clocking and baud rate

Both UART clock are individually controlled by means of the CCTRL_UART1 and CCTRL_UART2 bits of the SYS_CTRL_REG. Each UART contains a 12-bit down-counting scaler to generate the desired baud-rate. The scaler is clocked by the system clock and generates a UART tick each time it underflows. The scaler is reloaded with the value of the UART scaler reload register after each underflow. The resulting UART tick frequency should be 8 times the desired baud-rate.

Loopback

If the LOOPBACK bit in the UART control register is set, the UART will be in loop back mode. In this mode, the transmitter output is internally connected to the receiver input. It is then possible to perform loop back tests to verify operation of receiver, transmitter and associated software routines. In this mode, the outputs remain in the inactive state, in order to avoid sending out data.

FIFO

Both UARTs offer an input FIFO and an output FIFO. UART1 has a 16 bytes input and 16 bytes output FIFO. UART2 has a 4 bytes input and 4 bytes output FIFO.

For each FIFO a transmit and receive threshold can be programmed. An interrupt is raised when the receive FIFO content is higher than the receive threshold or the transmit FIFO content is lower than the transmit threshold. Both mechanisms can be disabled by setting the values of the thresholds to ‘0’. In this case an interrupt on transmit FIFO full or receive FIFO empty is raised.

An ageing mechanism raises an interrupt in the case where the receive FIFO threshold has not been reached, but data has been present in the FIFO for a given period of time. This mechanism avoids the need of polling the FIFO status when communicating on the UART. The ageing period is set by a counter value (0,16,32,64) that is counted to 0 with a signal generated from the GPS clock divided by 1024. If this period elapses, an interrupt is raised.

13.2 Register Details

UARTx_DATA_REG (Addr 0x80000070 / 0x80000080)

Field	Bits	Rst	Descriptions
RESERVED	31:8	undef	Reserved bits.
UART_DATA	7:0	0	Data: the register is shared for reading and writing data to the UART

UARTx_STAT_REG (Addr 0x80000074 / 0x80000084

Field	Bits	Rst	Description
RESERVED	31:30	undef	Reserved bits.
U_RX_FIFO_EMP	29	1	Receive FIFO empty
U_RX_FIFO_FULL	28	0	Receive FIFO full
U_RX_FIFO_CNT	27:24	0	Receive FIFO content
RESERVED	23:22	undef	Reserved bits.
U_TX_FIFO_EMP	21	1	Transmit FIFO empty
U_TX_FIFO_FULL	20	0	Transmit FIFO full
U_TX_FIFO_CNT	19:16	0	Transmit FIFO content
RESERVED	15:7	undef	Reserved bits.
T_TX_CH_EMPTY	8	0	Transmit channel empty. Is set when there are no characters in the FIFO, holding buffer and transmit shift register.
T_TX_BUFF_EMPTY	7	0	Transmit buffer empty. Is set when there are no characters in the FIFO or in the holding buffer, but there may be characters still being transmitted
U_FRAME_ERR	6	0	Framing error: framing error detected
U_PARITY_ERR	5	0	Parity error: parity error detected
U_RX_OVFL	4	0	Receive overflow
U_BRK	3	0	Break detected: indicates that a BREAK was received.
U_TX_QEMP	2	1	Transmit Queue Empty
U_TX_SR_EMP	1	1	Transmit Shift Register Empty
U_READY	0	0	Data Ready

UARTx_CTRL_REG (Addr 0x80000078 / 0x80000088)

Field	Bits	Rst	Description
RESERVED	31:30	undef	Reserved bits.
U_AGE	29:28	00	Ageing threshold (00= disabled, 01=16, 10=32, 11=64)
U_RX_TH	27:25	000	Receive threshold (00=disabled, 001=2, 010=4, 011=8, 100=10, 101=12, 110=14)
U_RX_PURGE	24	0	Receive purge
RESERVED	23:20	undef	Reserved bits.
U_TX_TH	19:17	000	Transmit threshold (00=disabled, 001=2, 010=4, 011=8, 100=10, 101=12, 110=14)
U_TX_PURGE	16	0	Transmit purge
RESERVED	15:8	undef	Reserved bits.
U_LOOPBACK	7	0	Loopback: if set, loop back mode will be enabled
RESERVED	6	undef	Reserved bits.
U_PARITY_EN	5	0	Parity Enable if set, enables parity generation and checking
U_PARITY_SEL	4	0	Parity Select selects parity polarity (0 = even parity, 1 = odd parity)
U_TX_IRQ_EN	3	0	Transmit IRQ enable
U_RX_IRQ_EN	2	0	Receive IRQ enable
U_TX_EN	1	0	Transmit enable
U_RX_EN	0	0	Receive enable

UARTx_USC_REG (Addr 0x8000007C / 0x8000008C)

Field	Bits	Rst	Description
RESERVED	31:12	undef	Reserved bits.
UART_USC	11:0	0	Clock divider factor. Divide SYS_CLK and generate the clock used by UARTx. It periodically counts from the programmed value down to 0.

14 GPIO

The GPIO shares 8 external signals to implement different functions. The 8 bits GPIO mode functionality is controlled by the SYS_GPIO_MODE bits of the SYS_CTRL_REG.

Modes 000 to 101 are functional modes, 110 and 111 are for debug and code profiling purpose.

14.1 GPIO Control Mode

MODE	GPIO[0]	GPIO[1]	GPIO[2]	GPIO[]3]	GPIO[4]	GPIO[5]	GPIO[6]	GPIO[7]
000	pio(0)	pio(1)	pio(2)	pio(3)	pio(4)	pio(5)	pio(6)	pio(7)
001	pio(0)	pio(1)	pio(2)	pio(3)	pio(4)	pio(5)	uart2_txd	uart2_rxd
010	pio(0)	spi_sl_sel(0)	spi_sl_sel(1)	spi_miso_i	spi_moso_o	spi_clk	uart2_txd	uart2_rxd
	pio(0)	spi_ssn_i	reserved	spis_mosi_i	spis_miso_o	spis_clk_in	uart2_txd	uart2_rxd
011	pio(0)	spi_sl_sel(0)	spi_sl_sel(1)	spi_miso_i	spi_moso_o	spi_clk	pps_valid	pps
	pio(0)	spi_ssn_i	reserved	spis_mosi_i	spis_miso_o	spis_clk_in	uart2_txd	uart2_rxd
100	pio(0)	pio(1)	acc_int	meas_int	pps_valid	pps	uart2_txd	uart2_rxd
101	pio(0)	pio(1)	CSN3	BRDYN	BEXCN	pps	uart2_txd	uart2_rxd
110	pio(0)	pio(1)	DMD_OUT	AHB_RDY	AHBRAM_CS	SCR_RAM_CS	ERROR	I_HIT
111	int0	int1	int2	int3	meas_int	Int_ack	psr_s	acc_int

14.2 GPIO Control Registers

Address	Name	Type	Function
0x800000A0	PIO_DATA_REG	Read/Write	I/O data
0x800000A4	PIO_DIR_REG	Read/write	I/O direction
0x800000A8	PIO_IRQ_REG	Read/write	I/0 interrupt control

If the GPIO lines are used as parallel IO (PIO), their functionality is defined by three registers. Two registers are associated with the operation of the I/O port; the combined I/O input/output register (PIO_DATA_REG), and I/O direction register (PIO_DIR_REG). When read, the PIO_DATA_REG register will return the current value of the I/O port; when written, the value will be driven on the port signals (if enabled as output). The direction register defines the direction for each individual port bit (0=input, 1=output).

The IO ports can also be used as interrupt inputs from external devices. A total of 2 interrupts can be generated and rerouted to interrupt lines 4 and 5 on the interrupt controller. The I/O port interrupt configuration register (PIO_IRQ_REG) defines which port should generate each interrupt and how it should be filtered.

14.3 Register Details

PIO_DATA_REG (Addr 0x800000A0)

Field	Bits	Rst	Description
RESERVED	31:8	undef	Reserved bits.
PIO_DATA	7:0	undef	Individual value set on or read form the PIO lines

PIO_DIR_REG (Addr 0x800000A4)

Field	Bits	Rst	Description
RESERVED	31:8	undef	Reserved bits.
PIO_DIR	7:0	0x0	Individually defines the direction of the PIO lines (0=input)

PIO_IRQ_REG (Addr 0x800000A8)

Field	Bits	Rst	Description
RESERVED	31:16	undef	Reserved bits.
PIO_IRQ5_EN	15	0	Enables IRQ5 source from PIO port
PIO_IRQ5_LEV	14	0	Selects between level (0) or edge (1) IRQ5 source
PIO_IRQ5_POL	13	0	Selects between active high (1) or active low (0) IRQ5 type
RESERVED	12:11	00	Reserved bits
PIO_IRQ5_SEL	10:8	000	Selects which of the PIO lines to use as source for IRQ5
PIO_IRQ4_EN	7	0	Enables IRQ4 source from PIO port
PIO_IRQ4_LEV	6	0	Selects between level (0) or edge (1) IRQ4 source
PIO_IRQ4_POL	5	0	Selects between active high (1) or active low (0) IRQ4 type
RESERVED	4:3	00	Reserved bits
PIO_IRQ4_SEL	2:0	000	Selects which of the PIO lines to use as source for IRQ4

15 SPI Master Interface

15.1 SPI Master Control Registers

Address	Name	Type	Function
0x800000D0	SPI_RXTX_REG	Read/Write	Transmitted/received data.
0x800000D4	SPI_CTRL_REG	Read/Write	Interface configuration.
0x800000D8	SPI_DIV_REG	Read/write	SPI interface clock frequency.
0x800000DC	SPI_SS_REG	Write	Select between 2 possible slaves.

A master SPI interface with 2 slave chip select is available as APB peripheral.

A SPI master is essentially a shift register that simultaneously shifts its contents out onto the SPI_MOSI_O (master-out slave-in) output line whilst shifting in received data from the SPI_MISO_I (master-in slave-out) input line. The transfer of data is directly controlled by the SPI clock. Data is transferred when the clock is active and held when it is not. This provides a simple synchronous communication interface to and from SPI slaves.

The SPI master must initiate and control all the SPI transfers between itself and the other SPI devices. It must generate the SPI interface clock and cannot respond to an external clock from another SPI device. Individual SPI slaves are selected via slave select lines. The APB_SPI provides 2 slave select lines so up to 2 slave devices may be directly connected. The slave select pins are directly driven so it is possible to use an external decoder and select up to 4 devices.

The APB_SPI generates the SPI data rate clock via a 16-bit clock divider, which pre-scales the APB PCLK down to the required clock frequency. The divider counts down to zero to produce half an SPI clock cycle. Thus a divider setting of 0 will produce a SPI_CLK clock of PCLK/2. This allows for a maximum division ratio of 2^{17}.

If required the APB_SPI can generate an interrupt at the end of a transmission sequence. The interrupt is active high and is cleared down by reading or writing to any APB_SPI register.

As there is no SPI specification to adhere to, the SPI module has been designed to allow easy interfacing to a variety of different SPI slaves. Thus the APB_SPI has several programmable features to cater for different SPI slave devices. The APB_SPI can be set to transmit and receive data on either edge of the SPI clock. This can be set separately for transmit and receive. The order in which data is transmitted and received can be set (MSB first or LSB first) and the physical polarity of the external SPI_CLK can be inverted. It can also initiate a transfer of between 1 and 32 bits of data.

During transmission, data is generated on the positive or negative edge of the SPI clock SCLK, according to the value of the C_SPI_CTRL_TX_NEG control bit. Whilst a transmission is occurring, data is simultaneously received on either the negative or the positive edge according to the value of the C_ SPI_CTRL_RX_NEG control bit. Note that it is expected that data shall be transmitted on one polarity of the clock edge and received on the other. This provides timing and noise immunity between the devices. However it is not particularly sensible to receive data on the positive edge and transmit on the negative edge, as this will effectively loose the first bit of data.

The APB_SPI uses 3 standard APB registers to implement the divider, the control, and the slave select registers. The data register, which contains the transmit/receive data, is part of the SPI shift register. All registers are read write capable, however they cannot be read whilst a transfer is taking place.

The entire design is fully synchronous and run from the single APB PCLK domain. The SPI clock is only a logical output as it is derived from the PCLK and nothing is actually clocked from it.

15.2 Register Details

SPI_RXTX_REG (Addr 0x800000D0)

Field	Bits	Rst	Description
SPI_RXTX	31:0	0	Data. This register is written with the data to be transmitted. After a transmission it contains the received data. Transmit data is justified to the LSB of this register: for a 16 bit transfer, the lower 16 bits will be sent.

SPI_CTRL_REG (Addr 0x800000D4)

Field	Bits	Rst	Description
RESERVED	31:12	undef	Reserved bits.
SPI_CTRL_LOOPBK	11	0	Self test loop back mode. When set, the transmitted data is inverted and fed back into the receive data.
SPI_CTRL_POL	10	0	Clock Polarity. When set, it inverts polarity of the SPI clock output.
SPI_CTRL_IE	9	0	Interrupt Enable. When set, this causes an interrupt to be generated upon completion of a transmit
SPI_CTRL_LSB	8	0	Least Significant Bit First.This affects the transmission order of the bits of C_SPI_RXTX. It also affects the ordering of received data. When set, it enables the transmission of data LSB first. i.e. C_SPI_RXTX (0) is transmitted first and the first bit of received data will be placed in C_SPI_RXTX (0). Receive data is also assembled in the reverse direction. 0 Enables the transmission of data MSB first. i.e. C_SPI_RXTX (0) is the LAST bit of data to be transmitted and received. The first bit will depend upon the length of the transmission as set by the C_SPI_CTRL_CHAR_LEN bits.
SPI_CTRL_CHAR_LEN	7:3	00000	Data length. This sets the size of the data word to be transmitted. ‘00000’ will transmit 1 bit, ‘00001’ will transmit 2 bits and so on.
SPI_CTRL_TX_NEG	2	0	Transmit on NEG edge. When set, this will cause the transmitter to generate data on each negative edge of the SPI clock
SPI_CTRL_RX_NEG	1	0	Receive on NEG edge. When set, this will cause the receiver to capture data on each negative edge of the SPI clock
SPI_CTRL_GO	0	0	Start transmission. Setting this bit will initiate a transmission sequence. The bit is automatically cleared at the end of the transmission sequence. Note that it is a condition of the original design that this should not be set when any of the other bits are being changed. Thus the control reg should be written with the appropriate control value and the go bit ='0'. Then it should be rewritten with the same control value and the go bit ='1' to start the transmission. If this condition is not met the behavior of the SPI may be unpredictable.

SPI_DIV_REG (Addr 0x800000D8)

Field	Bits	Rst	Description
SPI_DIV	15:0	0	Prescaler divide. This sets the division ratio of the prescaler used to generate the SPI interface clock. A value of 0x000 divides pclk by 2, 0x0001 by 4 and so on.

SPI_SS_REG (Addr 0x800000DC)

Field	Bits	Rst	Description
SPI_SS	1:0	0	Slave select. This drives the slave select output that selects appropriate slave device. The outputs are active low and so are the inverse of the register value.

16 SPI Slave

16.1 SPI Slave Control Registers

Address	Name	Type	Function
0x800000E0	SPISL_CTRL_REG	Write	General control of SPI slave
0x800000E4	SPISL_STATUS_REG	Read	Status of SPI slave
0x800000E8	SPISL_DATA_REG	Read/Write	Transmitted/received data

The SPI slave shares the IO pins of the SPI master by using the SYS_SPI_SLAVE register to control the direction of these pins.

The SPI slave device should receive the select, clock and data signals of the SPI master. When SPI test mode is activated by the TEST_SPI bit of the TEST_REG, data from the slave is routed back to the master so that a back-to-back test of the master and slave devices can be performed in the silicon.

The SPI slave has a 16 bytes receive and transmit FIFO and a fixed receive/transmit size of 8 bits (1 byte). It can generate an interrupt for each received byte, the interrupt being cleared by a register access. Both the receive and transmit FIFO have a programmable threshold level used to generate an interrupt.

For the receive FIFO an ageing mechanism is available that optionally generates an interrupt after a programmable ageing period, if data is present in the receive FIFO but the receive threshold level has not been reached. The ageing period can be selected between 16, 32 or 64 cycles of a tic signal that has the frequency of GPS_CLK divided by 1024.

16.2 SPI Slave Operation

Data can be written to the SPI slave up to the limit of the TX FIFO size. This data is transmitted to the master when a transfer occurs. Overfilling the TX FIFO will set the TX_FIFO error flag.

The master initiates a transfer by driving the slave select line low and toggles the clock to transfer data. New data is shifted into the slave as the TX data is returned to the master.

The transfer is terminated by the master driving the slave select line high where upon the current contents of the receive register are transferred to the RX FIFO.

If the interrupt generation is enabled, an interrupt is generated. It is cleared by accessing any slave register.

Data can be received up to the limit of the RX FIFO size. If the FIFO is not emptied before the next byte is received then the RX_FIFO error flag will be set to indicate a buffer overflow.

If the transmit buffer becomes empty and a transfer occurs the TX_FIFO error flag will be set to indicate a buffer underflow.

The SPI slave reuses the same output pins as the SPI master. The SYS_SPI_SLAVE selects the mode of the external signals, with 0 (default) being the master mode.

16.3 Register Details

SPISL_CTRL_REG (Addr 0x800000E0)

Field	Bits	Rst	Description
SPISL_TX_TRSH	14:12	000	Transmit threshold. 000 = disable, else generate an interrupt on transmitt fifo level = 4 (001), 6 (010), 8 (011), 10 (100), 12 (101), 14 (110), EMPTY (111)
SPISL_RX_TRSH	11:9	000	Receive threshold. 000 = disable, else generate an interrupt on receive fifo level = 4 (001), 6 (010), 8 (011), 10 (100), 12 (101), 14 (110), FULL (111)
SPISL_RXAGE	8:7	00	RX ageing value. 00 = disable, 01 = generate an interrupt when age count = 16, 10 when age count = 32 and 11 when age count = 64
SPISL_IRQ_IEN	6	0	If unset, generate an interrupt on every RX byte.
RESERVED	5	0	Reserved bit.
SPISL_CTRL_RX_PURGE	4	0	Clear the RX fifo.
SPISL_CTRL_TX_PURGE	3	0	Clear the TX fifo.
SPISL_CTRL_CTRL_POL	2	0	Sclk polarity
SPISL_CTRL_PHASE	1	0	Sclk Phase (early or late)
SPISL_CTRL_EN	0	0	Enable the core.

SPISL_STATUS_REG (Addr 0x800000E4)

Field	Bits	Rst	Description
SPISL_STATUS_INT	9	0	Interrupt flag - set by end of an SPI sequence, reset by accessing any register.
SPISL_STATUS_BUSY	8	0	Busy flag. Indicates that the slave is in the middle of a transfer.
SPISL_STATUS_TX_EMPTY	7	0	TX FIFO is empty.
SPISL_STATUS_TX_FULL	6	0	TX FIFO is full.
SPISL_STATUS_TX_ERROR	5	0	TX FIFO error. Set when attempting to read an empty FIFO or write a full one. Cleared by accessing the status register.
SPISL_STATUS_RX_EMPTY	4	1	RX FIFO is empty.
SPISL_STATUS_RX_FULL	3	0	RX FIFO is full.
SPISL_STATUS_RX_ERROR	2	0	RX FIFO error. Set when attempting to read an empty FIFO or write a full one. Cleared by accessing the status register.
SPISL_STATUS_RXLEVEL	1	0	RX FIFO level. Indicates the number of bytes in the FIFO.
SPISL_STATUS_TXLEVEL	0	0	TX FIFO level. Indicates the number of bytes in the FIFO.

SPISL_DATA_REG (Addr 0x800000E8)

Field	Bits	Rst	Description
SPISL_DATA	7:0	0	Data

17 DSU Port

17.1 DSU Control Registers

Address	Name	Type	Function
0x90000000	DSU_CTRL_REG	Read/Write	DSU main control register
0x90000010	AHB Break Address 1	Read/Write	AHB Address/Instruction to be detected
0x90000014	AHB Mask 1	Read/Write	Mask to apply on break value 1
0x90000018	AHB Break Address 2	Read/Write	AHB Address/Instruction to be detected
0x9000001C	AHB Mask 2	Read/Write	Mask to apply on break value 2

The LEON processor includes hardware debug support to aid software debugging on target hardware. The support is provided through two modules: a debug support unit (DSU) and a debug communication link (DCL) (see Figure 3). The DSU can put the processor in debug mode, allowing read/write access to all processor registers and cache memories. The debug communication link implements a simple read/write protocol and uses standard asynchronous UART communications (RS232C).

The optional trace buffer offered by the original Leon DSU is not implemented on the NJ1030.

The debug support unit is used to control the processor debug mode. The DSU is attached to the AHB bus as slave, occupying a 2 MByte address space. Through this address space, any AHB master can access the processor registers. The DSU control registers can be accessed at any time, while the processor registers and caches can only be accessed when the processor has entered debug mode. In debug mode, the processor pipeline is held and the processor state can be accessed by the DSU. Entering the debug mode can occur on the following events:

- executing a breakpoint instruction
- integer unit hardware breakpoint/watchpoint hit (trap 0xb)
- rising edge of the external break signal (DSUBRE)
- setting the break-now (BN) bit in the DSU control register
- a trap forcing the processor to enter error mode
- occurrence of any or a selection of traps as defined in the DSU control register
- after a single-step operation
- DSU breakpoint hit

The debug mode can only be entered when the debug support unit is enabled through an external pin (DSUEN). When the debug mode is entered, the following actions are taken:

- PC and nPC are saved in temporary registers (accessible by the debug unit)
- an output signal (DSUACT) is asserted to indicate the debug state
- the timer unit is (optionally) stopped to freeze the LEON timers and watchdog

The instruction that caused the processor to enter debug mode is not executed, and the processor state is kept unmodified. Execution is resumed by clearing the BN bit in the DSU control register or by de-asserting DSUEN. The timer unit will be re-enabled and execution will continue from the saved PC and nPC. Debug mode can also be entered after the processor has entered error mode (for instance when an application has terminated and halted the processor). The error mode can be reset and the processor restarted at any address.

17.2 External DSU signals

The DSU uses five external signals: DSUACT, DSUBRE, DSUEN, DSURX and DSUTX. They are used as follows:

- DSUACT - DSU active (output). This active high output is asserted when the processor is in debug mode and controlled by the DSU.
- DSUBRE - DSU break enable. A low-to-high transition on this active high input will generate break condition and put the processor in debug mode. After a low-to-high transition is detected, up to four instructions will be executed before debug mode is entered.
- DSUEN - DSU enable (input). The active high input enables the DSU unit. If de-asserted, the DSU trace buffer will continue to operate but the processor will not enter debug mode.
- DSURX - DSU receiver (input). This active high input provides the data to the DSU communication link receiver.
- DSUTX - DSU transmitter (output). This active high output provides the output from the DSU communication link transmitter.

17.3 DSU Breakpoint Registers

The DSU offers two breakpoints registers for matching either a AHB address or an executed processor instructions. A breakpoint hit is typically used to put the processor in debug mode. A mask register is associated to each breakpoint register allowing breaking on a block of the break value, as specified by the bits that are set to ‘1’ in the mask. To break on executed instructions, the EX bit should be set in the BADDRx register. To break on AHB load or store accesses, the LD and/or ST bits must be set in the MASK register.

17.4 DSU UART

The DSU communication link consists of a UART connected to the AHB bus as a master. A simple communication protocol is supported to transmit access parameters and data. A link command consist of a control byte, followed by a 32-bit address, followed by optional write data. If the LR bit in the DSU control register is set, a response byte will be sent after each AHB transfer. If the LR bit is not set, a write access does not return any response, while a read access only returns the read data. Data is sent on 8-bit basis.

Through the communication link, a read or write transfer can be generated to any address on the AHB bus. A response byte can optionally be sent when the processor goes from execution mode to debug mode. Block transfers can be performed by setting the length field to n-1, where n denotes the number of transferred words. For write accesses, the control byte and address is sent once, followed by the number of data words to be written. The address is automatically incremented after each data word. For read accesses, the control byte and address is sent once and the corresponding number of data words is returned.

The UART interface automatically detects the baudrate once the DSU monitor program is launched on the host.

The NJ1030 has a dedicated port for the DSU UART. However, when the DSU_MUX pin is high, the DSU TX and RX signal are connected to the TX and RX signals of UART1. This allows the access to the DSU port for debug or SW update purposes when no dedicated RS-232 buffers is present on the PCB for the DSU port.

17.5 Common operations

17.5.1 Instruction breakpoints

Instruction breakpoints can be inserted by writing the breakpoint instruction (ta 1) to the desired memory address (software breakpoint) or using any of the four integer unit hardware breakpoints. Since cache snooping is only done on the data cache, the instruction cache must be flushed after the insertion or removal of software breakpoints. To minimize the influence on execution, it is enough to clear the corresponding instruction cache tag valid bit (which is accessible through the DSU).

The two DSU hardware breakpoints should not be used for software debugging since there is a 4-instruction delay from the breakpoint hit before the processor enters the debug mode.

17.5.2 Single stepping

By setting the SS bit and clearing the BN bit in the DSU control register, the processor will resume execution for one instruction and then automatically return to debug mode.

17.5.3 Booting from DSU

By asserting DSUEN and DSUBRE at reset time, the processor will directly enter debug mode without executing any instruction. The system can then be initialised from the communication link, and applications can be downloaded and debugged. Additionally, external (flash) proms for standalone booting can be re-programmed.

17.6 DSU Register Details

DSU_CTRL_REG (Addr 0x90000000)

Field	Bits	Rst	Description
RESERVED	31:20	undef	Unused bits.
RE	19	undef	Reset error mode (RE) - if set, will clear the error mode in the processor.
DR	18	undef	Debug mode response (DR) - if set, the DSU communication link will send a response word when the processor enters debug mode.
LR	17	undef	Link response (LR) - If set, the DSU communication link will send a response word after AHB transfer.
SS	16	undef	Single step (SS) - if set, the processor will execute one instruction and the return to debug mode.
PE	15	undef	Processor error mode (PE) - returns ‘1’ on read when processor is in error mode, else ‘0’.
EE	14	undef	EE - value of the external DSUEN signal (read-only).
EB	13	undef	EB - value of the external DSUBRE signal (read-only).
DM	12	undef	Debug mode (DM). Indicates when the processor has entered debug mode (read-only).
RESERVED	11	undef	Unused bits
BZ	10	undef	Break on error traps (BZ) - if set, will force the processor into debug

			mode on all *except* the following traps: priviledged_instruction, fpu_disabled, window_overflow, window_underflow, asynchronous_interrupt, ticc_trap.
BX	9	undef	Break on trap (BX) - if set, will force the processor into debug mode when any trap occurs.
BB	8	undef	Break on DSU breakpoint (BD) - if set, will force the processor to debug mode when a DSU breakpoint is hit.
BN	7	undef	Break now (BN) -Force processor into debug mode. If cleared, the processor will resume execution.
BS	6	undef	Break on S/W breakpoint (BS) - if set, debug mode will be forced when a breakpoint instruction (ta 1) is executed.
BW	5	undef	Break on IU watchpoint - if set, debug mode will be forced on an IU watchpoint (trap 0xb).
BE	4	undef	Break on error (BE) - if set, will force the processor to debug mode when the processor would have entered error condition (trap in trap).
FT	3	undef	Freeze timers (FT) - if set, the scaler in the LEON timer unit will be stopped during debug mode to preserve the time for the software application.
BT	2	undef	Break on trace (BT) - if set, will generate a DSU break condition on trace freeze.
DM	1	undef	Delay counter mode (DM). In mixed tracing mode, setting this bit will cause the delay counter to decrement on AHB traces. If reset, the delay counter will decrement on instruction traces.
RESERVED	0	undef	Unused bit

DSU_BADDR1/2_REG (Addr 0x90000010/18)

Field	Bits	Rst	Description
BADDR	31:2	undef	AHB Address or instruction to be matched
RESERVED	1	undef	
EX	0	undef	If ‘1’ then a break on executed instruction is generated.

DSU_BMASK1/2_REG (Addr 0x90000014/1C)

Field	Bits	Rst	Description
BMASK	31:2	undef	AHB Address or instruction to be matched
LD	1	undef	If ‘1’ then a break on load access is generated.
ST	0	undef	If ‘1’ then a break on store access is generated

18 ADC

18.1 ADC Control Register

Address	Name	Type	Function
0x800000B0	ADC_CTRL_REG	Read/Write	ADC value and enable

An 8 bit ADC with a sampling rate of 1kHz is available. Combined with an on-chip temperature sensor with a resolution of 0.75°C, the ADC achieves temperature compensation in the range between -60 to 132°C. This eliminates the need of an expensive TCXO in many GPS applications.

Seven analog inputs are multiplexed to the ADC: the usable range is between 0V and 1.2V (higher voltages are treated as 1.2V). This allows the seamless integration of other sensors like a compass or a gyroscope.

The temperature sensor provides an 8 bit value proportional to the chip temperature that can be then used by the SW to be translated into a temperature value between –60°C to 132°C with a 0.75°C resolution. The temperature read mechanism is based on a 10 bit ADC_CTRL_REG register which entails the 8 bit ADC_VALUE, the ADC_EN bit and the ADC_VALID bit.

The ADC is built with a successive approximation register approach. The ADC_VALUE is converted into an analog signal by a DAC; this analog signal is compared with the signal to be converted into digital. The comparator delivers a COMP ='1' value if the incoming analog signal is higher than the one produced by the DAC. The procedure is repeated for each bit of the ADC_VALUE, starting from the most significant bit. The procedure is implemented by a finite state machine that is started by the ADC_EN signal and stops after asserting an ADC_VALID signal.

18.2 Temperature Sensor Utilization Procedure

From the CPU perspective:

The CPU writes a '1' in the ADC_EN bit
The CPU polls the ADC_CTRL_REG until a high ADC_VALID bit is found

When the enable bit is activated the ADC controlling logic behaves in the following way:

The ADC_VALUE is initialised to '0'
Then from the MSB down to the LSB of ADC_VALUE

- the bit is set to '1'
- if COMP ='1' then the bit is maintained at '1'
- else it is cleared

The ADC_VALID bit is set
A CPU read operation with a high ADC_VALID, clears the ADC_EN bit so that the temperature sensor is deactivated.

The ADC has 7 external analog inputs that are selected by the value of the SYS_ADC_SEL bits of the SYS_CTRL_REG. Default value is '000' and selects the Tsens input, while '001' to '111' select ANALOG_IN[0] to ANALOG_IN[6].

Note that AIN[0..6] are high impedance inputs, the voltage should be DC or vary less than 1 LSB/ms (1 LSB = 4.85 mV).

18.3 Register Details

ADC_CTRL_REG (Addr 0x800000B0)

Field	Bits	Rst	Description
ADC_EN	9	0	ADC Enable. Set by the CPU to initiate a conversion. Autonomously cleared at the end of a conversion when ADC_VALID is set.
ADC_VALID	8	0	End of conversion. Set by the ADC when the conversion is terminated. Once set, it can be cleared by a read from the ADC_CTRL_REG.
ADC_VALUE	7:0	0x00	ADC conversion value.

19 Power Supplies

The NJ1030 has 4 separate pad rings and one core power supply in order to offer different interfacing options while minimizing power consumption.

The pad ring power supplies are:

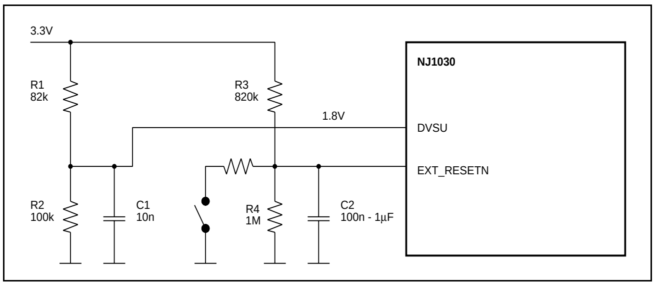
- DVSU: 1.6V to 2V. Power supply for the power up unit and the portion of pad ring with EXT_WAKE, CLK_SOURCE, EXT_RESET, EXT_VREGN and EXT_EN. This power supply must be present in order to wake up the system from a sleep mode with an EXT_WAKE.
- DVDD 1.8V to 3.6V. Supply (noisy) for pad ring and voltage regulator (to generate LVDD). The level of this voltage is determined by the external digital components used in the system. This power supply should be monitored by the power supervisor.
- AVDD 1.8V to 3.6V. Supply (quiet) for on chip analog blocks (excl. Voltage Regulator). AVDD should possibly be derived from DVDD via a RC filter.
- TVDD: 1.8V to 3.6V separate power supply for IF interface to the RF front end.

The core power supply LVDD can be generated on chip by the on chip voltage regulator, and its level is set by the voltage present on the VBF pin. Otherwise it can be generated externally by disabling the on chip voltage regulator by setting the EXT_VREGN to 0 and override the voltage on the LVDD pins.

VBAT can have a value between 1.1V and 2V obtained from a battery, an accumulator or a capacitor. A VBAT voltage lower than 1.2V raises the LOW_BATTERY bit in the SYS_CTRL_REG.

In a system with a DVDD > 1.8V the following circuit can be used to generate the 1.8V power supply needed for the DVSU domain and its related signals. The shown circuit refers to the situation where DVDD is 3.3V.

Figure 13: DVSU generation circuit from 3.3V DVDD



19.1 Power Supervisor

- **VRTC1:** When the main DVDD or LVDD , VRTC1 (for the RTC clock logic and NVRAM block) switches from LVDD to VBAT.
- **VRTC2**: When the main AVDD is down, VRTC2 (for the RTC oscillator) switches from AVDD to VBAT.

VRTC1 and VRTC2 are internal power supplies. VRTC1 is the power supply of the RTC logic and of the battery backed up memory. VRTC2 is the power supply of the RTC oscillator.

20 Reset Sequence

The System Reset Sequence can be asynchronously initiated by an EXT_RESET low, a WATCHDOG low (if not disabled by the WDG_RESET_DISABLE bit in the SYS_CTRL_REG) or a POWER_OK low signal given by the power supervisor.

The recovery from a reset goes through two sequential phases. In the first one the validity of the SYS_CLK signal is verified by a 16 bit counter that must overflow.

Once the clock has been verified, a second counter (6 bits) is used to keep the internal reset low. When this counter overflows the system reset is raised (synchronously) and the boot procedure is initiated by the CPU.

A reset procedure initiated by a POWER_OK low signal goes through the two phases, while in the case where a EXT_RESET or a WATCHDOG started the procedure, only the second reset phase is entered.

21 Operating Modes

The system has 3 main operating modes that can be set under control of the CPU: FULLY ACTIVE, STAND_BY and SLEEP.

21.1 Fully Active Mode

This is the normal operating mode at the selected clock frequency. For the individual blocks the appropriate clocking mode is selected by the CCTRL_REG. Power supplies are in the following status.

Power Supply	Status
DVSU	On
DVDD	On
LVDD	On, internal or external
TVDD	On
AVDD	On
VRTC1	LVDD
VRTC2	AVDD

21.2 Stand By

The NJ1030 power supplies are in the same status as the fully active mode, while the CPU stops its own clock when no further activity is needed. The clocking of the peripherals is set according to their activity. The CPU clock is reactivated at the occurrence of any of the interrupts.

STAND_BY sequence:

1. the CPU sets the DMD_EN bit for the CPU_CLK and the MC_CLK.

2. After the memory controller has finished any pending transfer operation it releases its clock demand signal, so that the clock of the CPU and the MC clocks are shut down.

3. As soon as an interrupt occurs, the DMD_EN bits of the CPU and the memory controller are cleared so that program execution is resumed. Program execution restarts form the point where sleep mode was entered.

This mode can for instance be used by the GPS software stack where the CPU puts itself in sleep mode after the execution of the tracking loop service routine.

21.3 Sleep Mode

Under control of the CPU (setting the SYS_GO_SLEEP bit of SYS_CTRL_REG), the internal analog blocks are deactivated. If enabled, internal LVDD generation is shut down and the external EXT_EN signal goes to 0. If used, the on chip oscillator is switched off while an external clock source is used, then the clock signal path is switched to the internal source.

The EXT_EN signal can be used to switch off any external device or clock source, according to the system design requirements.

Operation is resumed with a normal reset procedure, triggered by either a RTC_WAKE, an EXT_WAKE or a system reset. Resuming operation doesn't require the presence of a clock.

If the system isn't in SLEEP MODE, an RTC_WAKE or EXT_WAKE events will not send the system in reset but can generate an interrupt (see Interrupt Controller, chapter 7).

During SLEEP MODE, power supplies are in the following status.

Power Supply	Status
DVSU	On, necessary to recover
DVDD	On
LVDD	Off when internally generated, otherwise it depends on the ext. source
TVDD	Depends on external source
AVDD	Depends on external source
VRTC1	VBAT
VRTC2	VBAT

When LVDD is off and DVDD is still present, the output and bidirectional signal pins are put into a pre-defined mode, as defined in the pinout table.

Power Supplies

DVSU	DVDD	AVDD	LVDD	Status
0	0	0	0	Power off
0	0	0	1	**Damage - forbidden**
0	0	1	0	Invalid
0	0	1	1	**Damage - forbidden**
0	1	0	0	Invalid - stable
0	1	0	1	Invalid - instable
0	1	1	0	Invalid - stable
0	1	1	1	Invalid - instable
1	0	0	0	Sleep mode
1	0	0	1	**Damage - forbidden**
1	0	1	0	Sleep mode
1	0	1	1	**Damage - forbidden**
1	1	0	0	Sleep mode
1	1	0	1	Invalid - instable
1	1	1	0	Sleep mode
1	1	1	1	Operating or stand by

21.4 EXT_EN functionality

Some restriction apply to the generation of EXT_EN. In order to be able to use the EXT_EN pad to activate/deactivate external components, the followings rules must be followed:

- In the case where LVDD is generated on chip:
  - DVDD must be always ON and LVDD is automatically switched off in sleep mode by the internal voltage regulator.
  - At power-up or after a wake-up, the on-chip voltage regulator is activated. LVDD is generated which allows the propagation of EXT_EN to the external components.
- In the case where LVDD is generated externally
  - LVDD must be always ON.
  - DVDD must be always ON since the situation where LVDD≠0 and DVDD=0 is not allowed in the system (see chapter 20.3)

In both cases, EXT_EN can be used to switch on and off external components in the DVDD domain (or other power domains) such as a TCXO. During sleep mode, this feature limits the power consumption on DVDD to static currents.

22 Power Up Unit

A fully asynchronous power up unit, in the DVSU power supply domain implements the following functions:

- Provide an asynchronous mechanism to resume from the SLEEP mode, which is entered by setting of the GO_SLEEP bit by the CPU. Exit of this mode is possible from a status without clock and LVDD power by means of the RTC_WAKE or the EXT_WAKE or an EXT_RESET.
- Generate the enable signal for the analog blocks (voltage regulator, power supervisor and bandgap reference). The enable deactivates these blocks when the system enters SLEEP_MODE.
- Generate the enable signal for the main oscillator that is always disabled when an external clock source is used and is disabled in SLEEP_MODE if the on chip oscillator is used.
- Generate the select signal for the main oscillator so that when the system is in SLEEP_MODE the internal clock is selected, otherwise the clock source is selected as a function of the CLK_SOURCE.

23 Analog Signal Connection

23.1 Voltage Regulator Feedback Voltage (VFB)

When the internal voltage regulator is enabled, VFB is used to set its output voltage. A resistive divider (R1 and R2 in picture 13) must be placed between LVDD and VSS.

Vh = Vref*(R1+R2)/R2

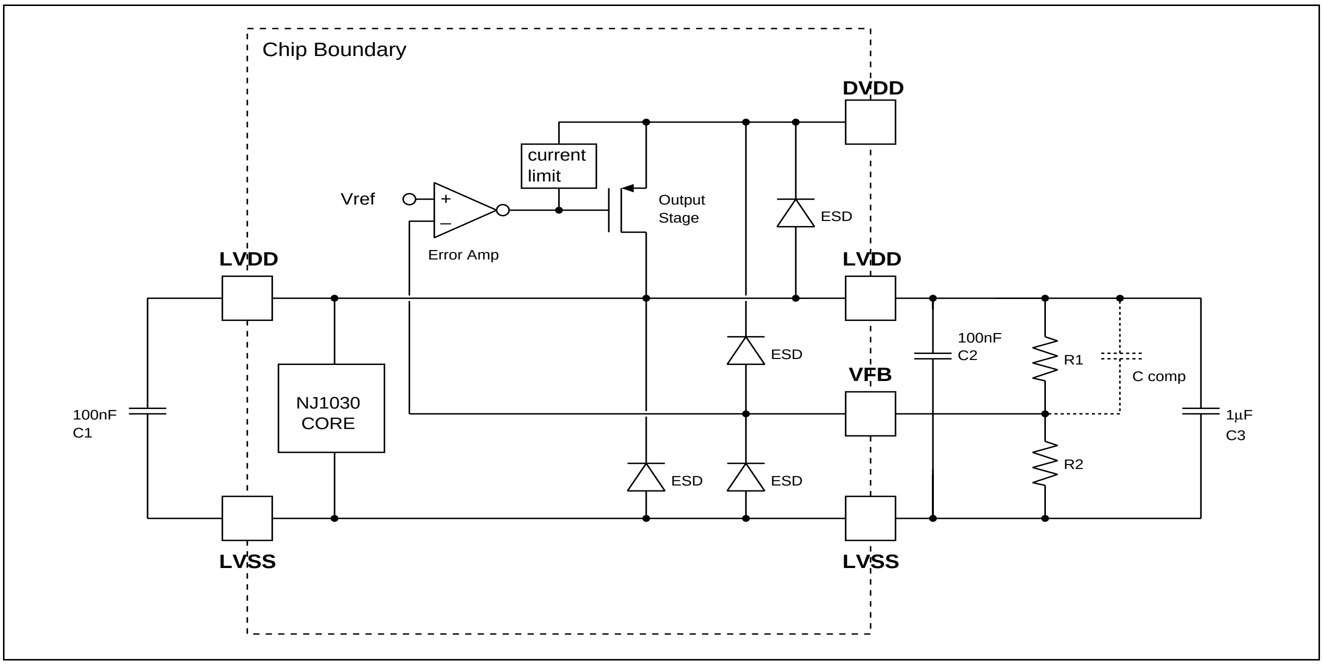
where R1 is the top resistor and R2 the bottom one. The sum of R1 and R2 should be in the range of 10 kΩ to 100kΩ. Higher values might make the regulator more sensitive to noise on LVDD, smaller values will result in high load current. C_{comp} = 1uF may be necessary if R1 is higher than 20kΩ or if high stray capacitance on pin VFB is anticipated.

A capacitor should be placed between LVDD and VFB if the resistor's values are high, in order to avoid lowpass behavior in the feedback circuit (1nF recommended).

A capacitor of at least 22nF (recommended 100nF) must be connected between LVDD and LVSS on both sides of the chip (C1 and C2), in addition to the usual 1uF (C3) decoupling capacitor. C3 may be increased if desired to reduce noise on LVDD.

When an external regulator is used, the internal voltage regulator must be disabled (pin EXT_VREGN). VFB may be connected either to VSS, LVDD or any other voltage in between. VFB must not be left open.

Figure 14. VFB connection diagram



23.2 Voltage Supervisor Input (VSI)

This pin is used to set the trip voltage of the power supervisor. A resistive divider (R1 and R2) must be placed between the voltage that must be monitored and VSS.

If xVDD is the power supply to be supervised:

- xVDD > Vh → the power level is considered as OK.
- xVDD < Vl → the power level is considered as NOT OK and the chip will be reset.

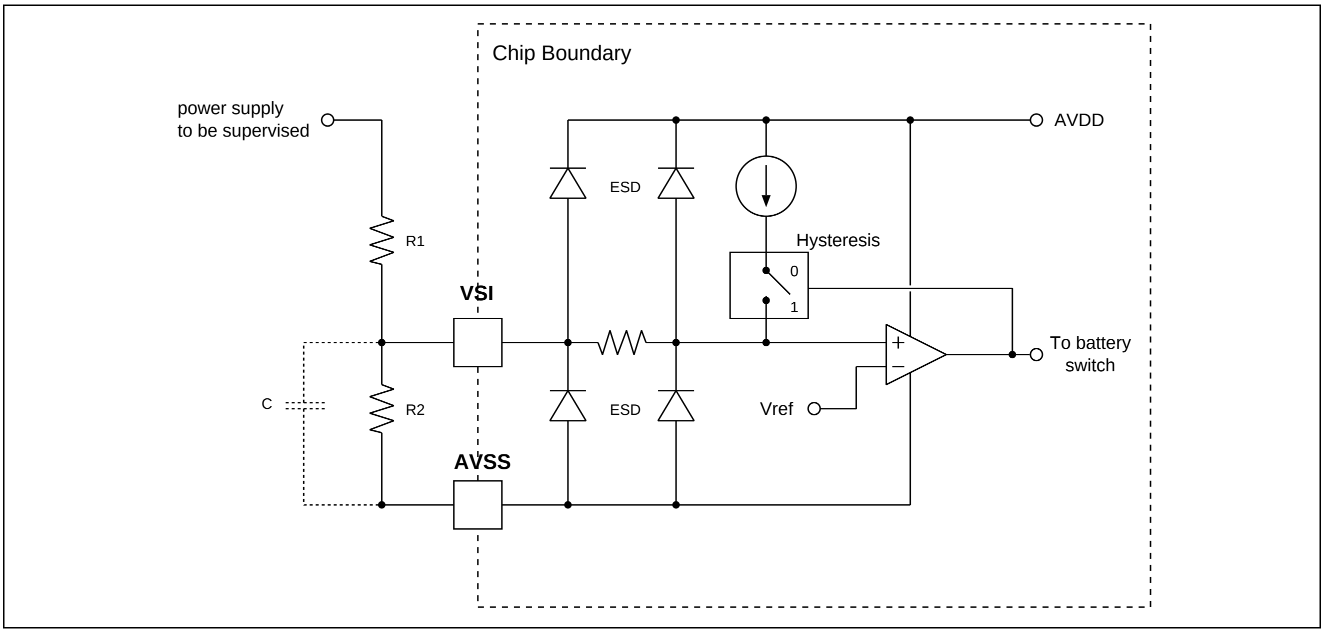
$Vh = Vref*(R1+R2)/R2$

$Vl = Vref*(R1+R2)/R2 – I_{HYST}*R1$

Where R1 is the top resistor and R2 the bottom one and with I_{HYST} = 1.6 µA typically.

It is recommended to monitor DVDD or the earliest voltage that will fall, e.g. the supply of the on-board regulators.

Figure 15. VSI connection diagram



23.3 Main Oscillator (MXI and MXO).

MXI and MXO are the input and output pins of the main oscillator. This oscillator can be used in 4 different ways:

- Quartz crystal and load capacitors on MXI and MXO, oscillator enabled (by the CLK_SOURCE pin).
- TCXO on MXI, internal oscillator used as buffer. Oscillator must be enabled. MXO floating.
- Digital clock on MXI, internal oscillator disabled.

The max frequency is 35 MHz.

23.4 RTC Oscillator (RXI and RXO).

The Real Time Clock Oscillator is a normal RTC oscillator. It should be connected to a quartz crystal and load capacitors on RXI and RXO. A 10MΩ resistor must be placed between RXI and RXO (DC biasing).

23.5 Voltage Reference Output (VREF)

This is the voltage reference output. A 10nF capacitor must be placed between VREF and VSS.

The voltage reference can source about 200µA. It cannot sink any current (no pulldown). It can be used externally, but no noise should be introduced on this pin.

Figure 16. Vref connection diagram

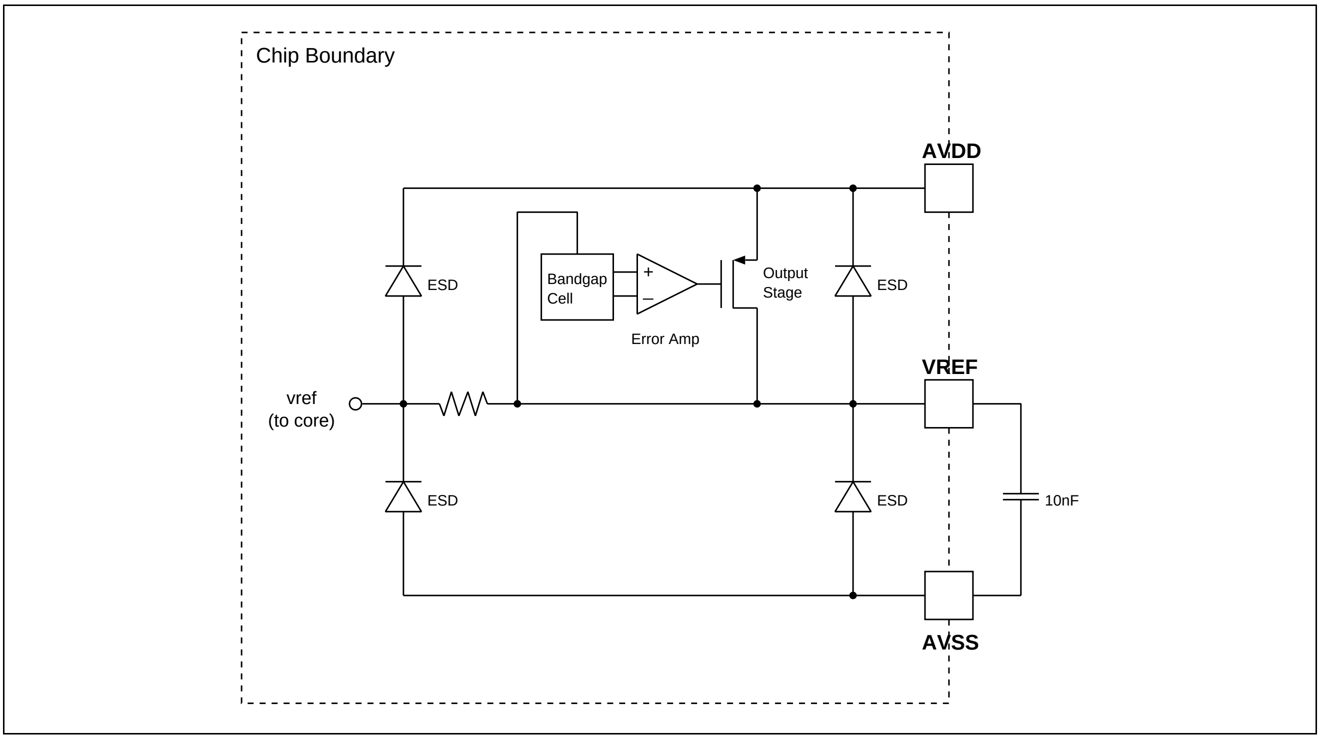


Figure 17.GPS Baseband example – 1.8 V Digital interface power supply

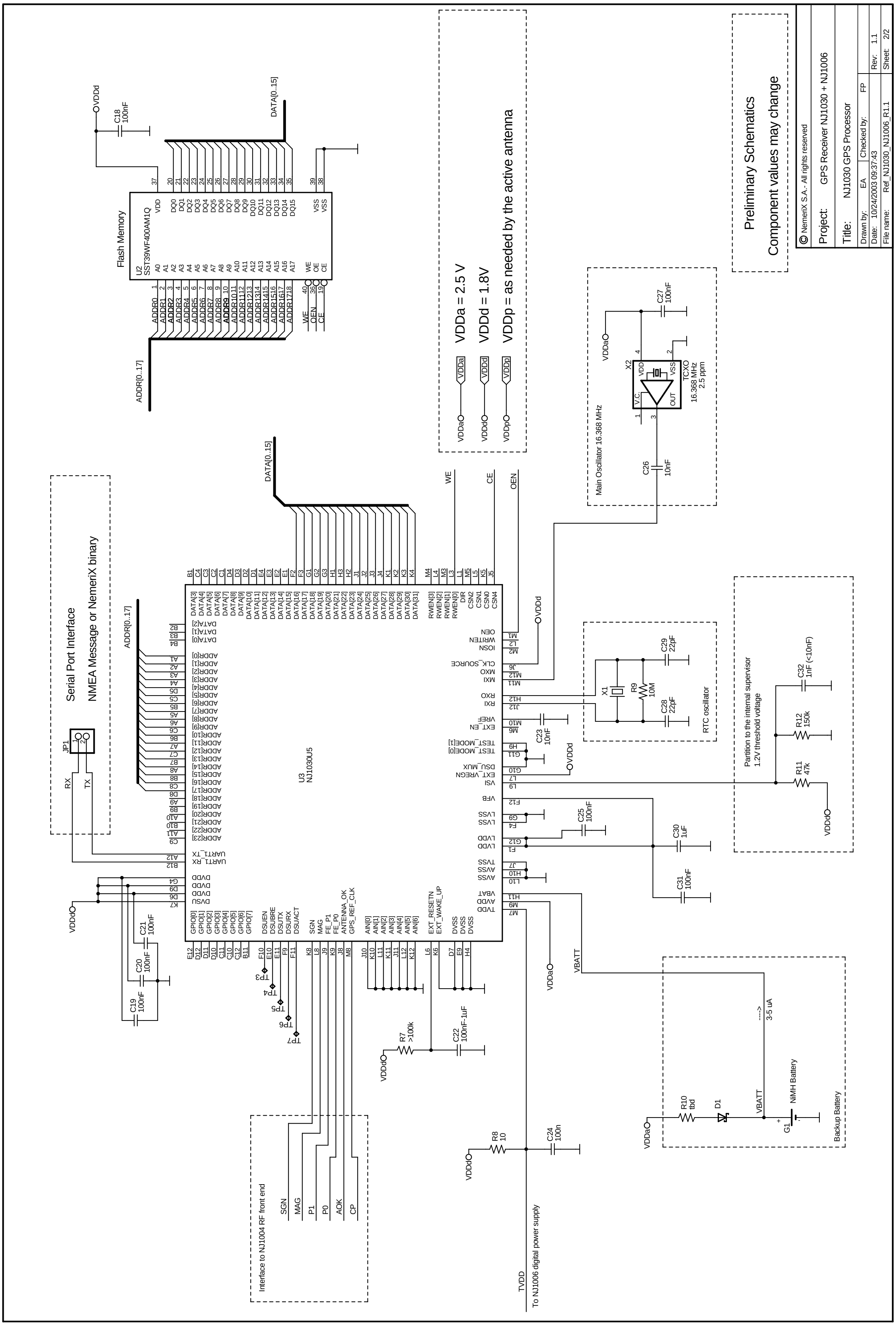


Figure 18. GPS Baseband example – 3.3 V Digital interface power supply

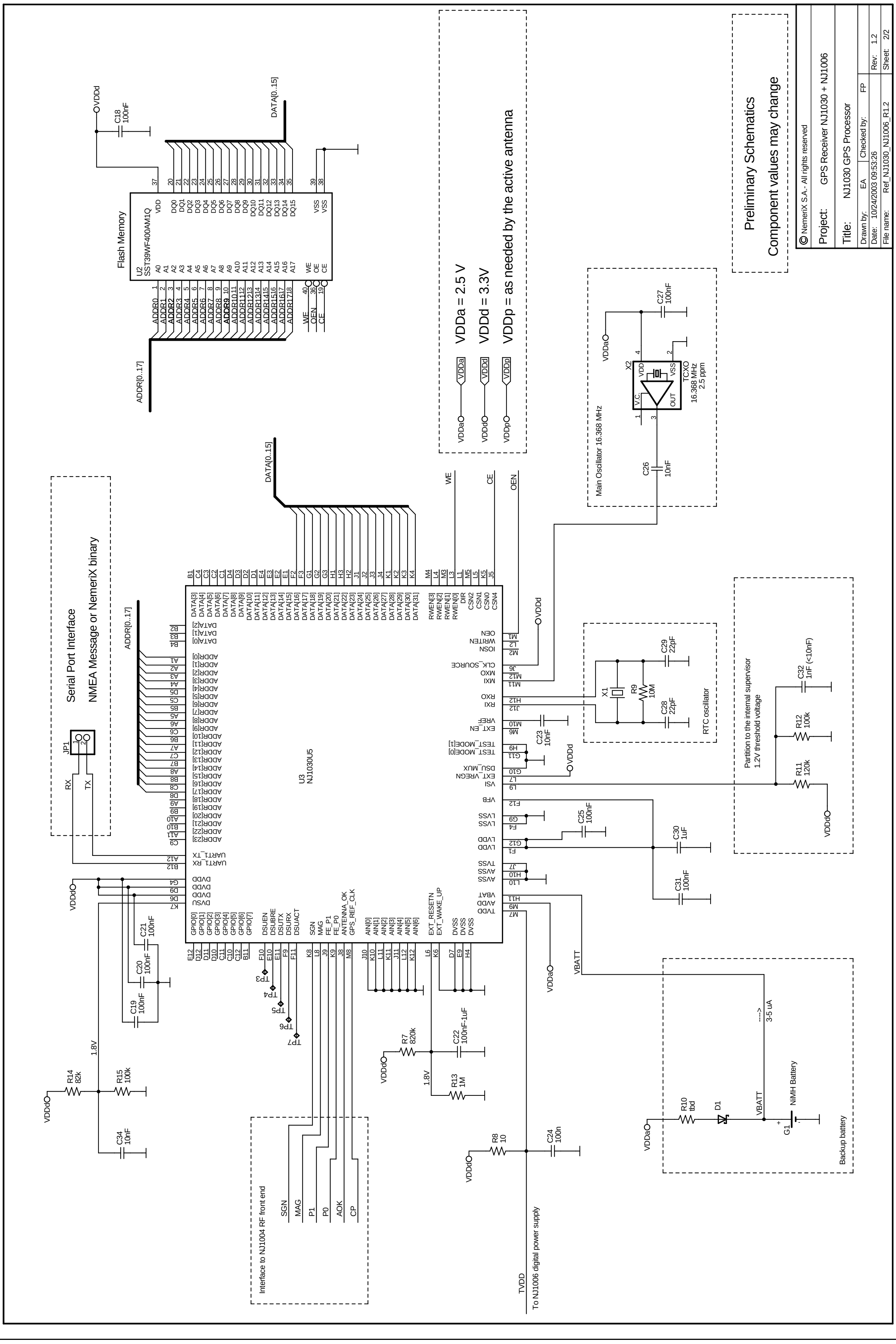


Figure 19. GPS Front End example – 1.8 V or 3.3 V Digital interface power supply

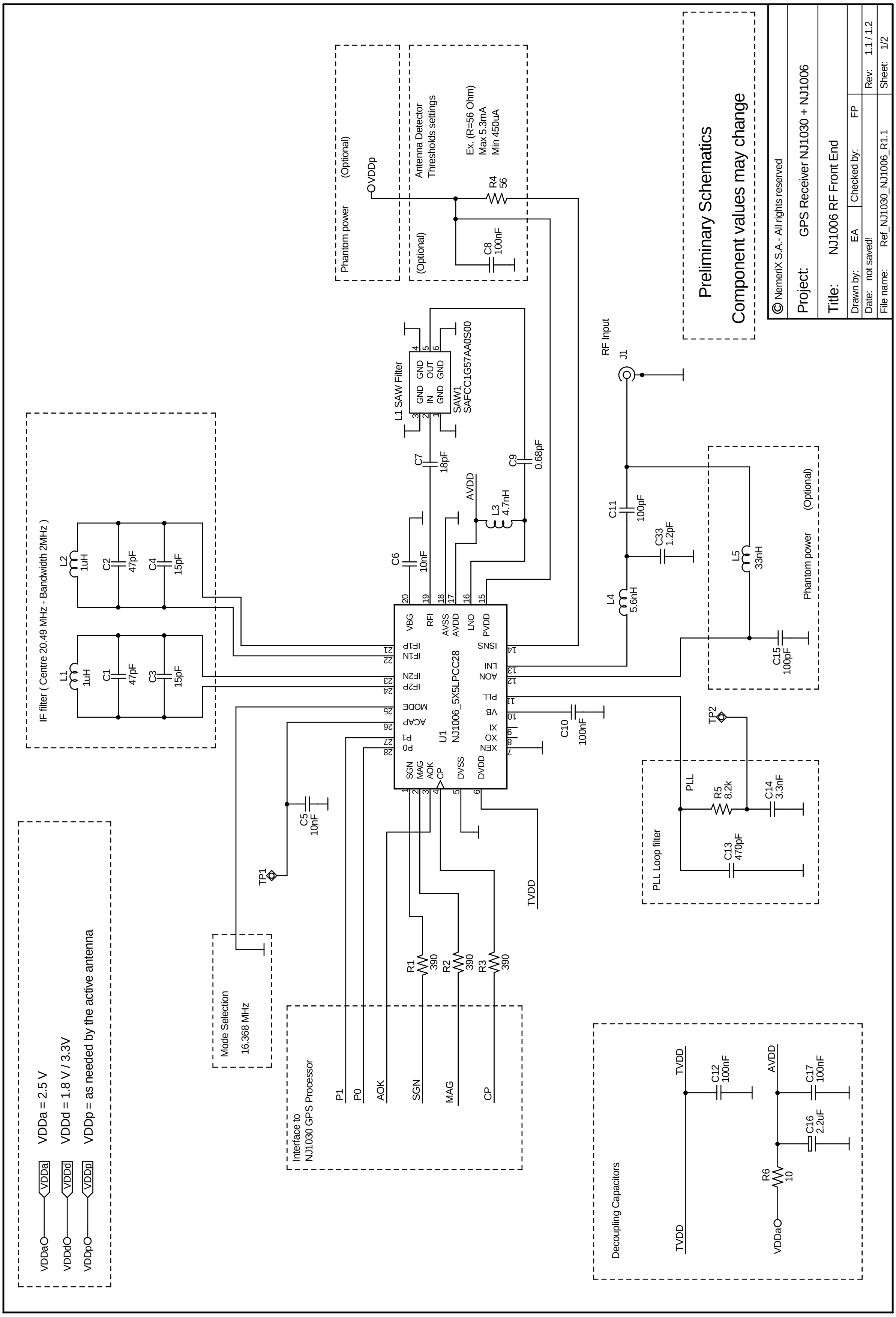
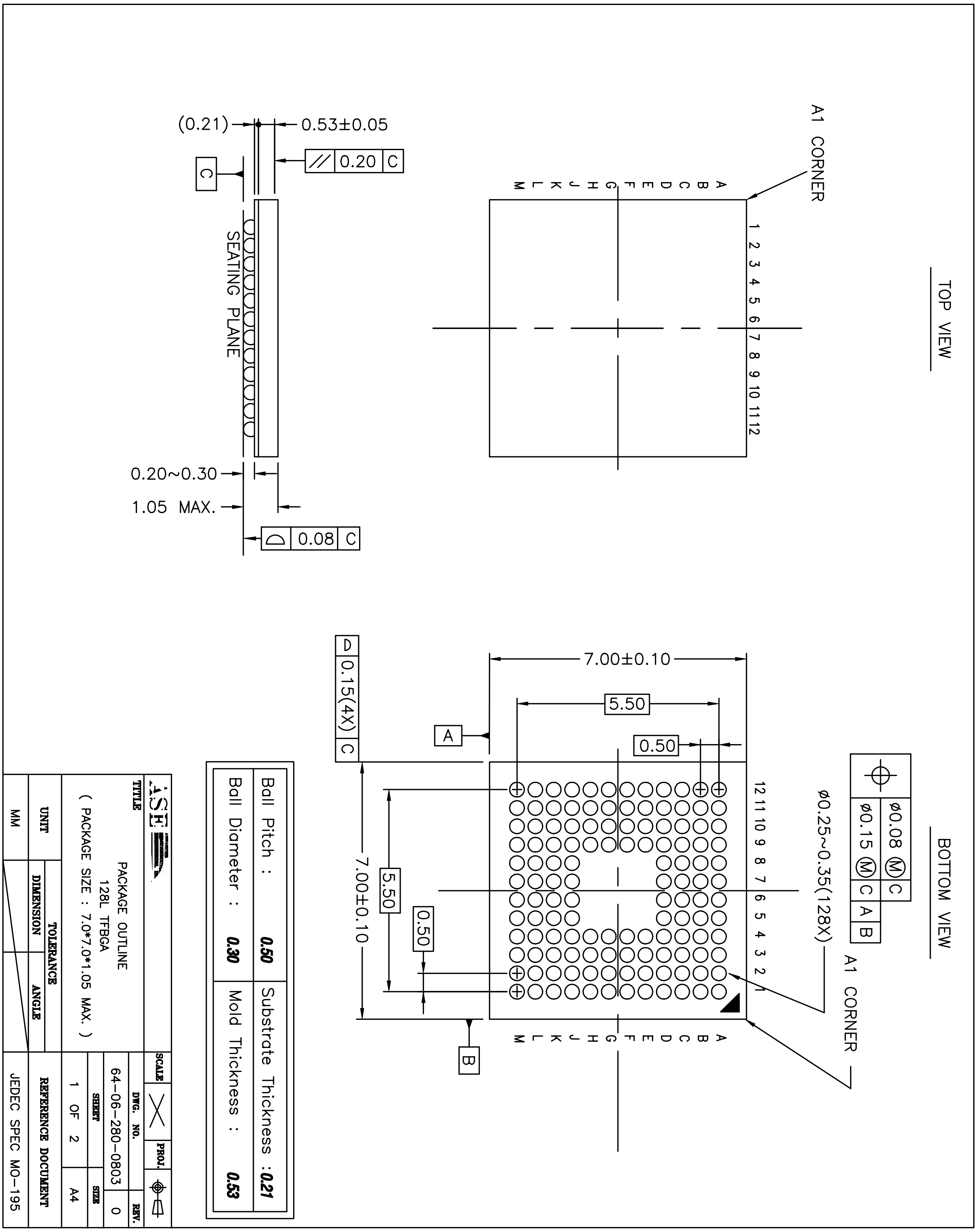


Figure 21. Micro BGA package specifications



Notes:

Ordering information

Part	Description
NJ1030U5	GPS Baseband Processor

Related products

Part	Description
NJ1006	GPS RF Front-End
NP1016	GPS correlator IP Core
DK1030	Software Development Kit

NemeriX SA
Headquarters
Stabile Gerre 2000
PO Box 425
6928 Manno
Switzerland

Phone +41 91 612 4700
Fax +41 91 612 4701

email sales@nemerix.com

NemeriX SA
United State Sales Office
5901 London Court
Dallas, TX 75252
USA

Phone +1 214-763-5760

email sales.usa@nemerix.com

LEGAL NOTICE:

NEMERIX PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR IN APPLICATIONS WHICH INVOLVE POTENTIAL RISK OF DEATH, PERSONAL INJURY OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE IN CASE OF FAILURE OR MALFUNCTION OF THE PRODUCT.