



UNOFFICIAL

P2X8C4M64P 8-Core Realtime Processor Preliminary Shortform Data

2012082300

The P2X8C4M64P is the first member of the Parallax P2 family of realtime controllers containing 8 identical 32-bit processors called "cogs", which connect to a common "hub". The hub provides a shared RAM, a CORDIC math solver, and common system resources.

The architecture supports up to 64 smart I/O pins, each capable of many autonomous analog and digital functions such as UARTS, PWM, A/D, D/A etc.

COG x8

- 200MHz 32-bit CPU - (Overclocks to 360MHz)
Equal access to all I/O pins, plus four fast DAC output channels
512 longs of dual-port register RAM for code and fast variables
512 longs of dual-port lookup RAM for code, streamer lookup, and variables
Ability to execute code directly from register RAM, lookup RAM, and hub RAM
~350 unique instructions for math, logic, timing, and control operations
2-clock execution for all math and logic instructions, including 16 x 16 multiply
6-clock custom-bytecode executor for interpreted languages
Stream hub RAM and/or lookup RAM to DACs and pins, also pins to hub RAM
Live colorspace conversion using a 3 x 3 matrix with 8-bit signed/unsigned coefficients
Pixel blending instructions for 8:8:8 data
16 unique event trackers that can be polled and waited upon
3 prioritized interrupts that trigger on selectable events
Hidden debug interrupt for single-stepping, breakpoint, and polling

HUB

- 512kB of contiguous RAM in a 20-bit address space
32-bits-per-clock sequential read/write for all cogs, simultaneously
readable and writable as bytes, words, or longs
last 16KB of RAM also appears at end of 1MB map and is write-protectable
16 semaphore bits with atomic read-modify-write operations
32-bit free-running counter, increments every clock
Mechanisms for starting, polling, and stopping cogs
16KB boot ROM loads into last 16KB of hub RAM on boot-up
SPI Flash and SD boot loader for automatic startup
Serial loader for startup from host + Monitor + TAQOZ interactive interpreter

CORDIC SOLVER

- 32-bit, pipelined CORDIC solver with scale-factor correction
32 x 32 unsigned multiply and 64 / 32 unsigned divide
64 -> 32 square root
Rotate (X32,Y32) by Theta32 -> (X32,Y32)
Convert between Polar (Rho32,Theta32) -> Cartesian (X32,Y32)
32 -> 5:27 unsigned-to-logarithm and 5:27 -> 32 logarithm-to-unsigned
CORDIC operations can start every 1/2/4/8/16 clocks with results in 55 clocks

PRNG Pseudo Random Number Generator

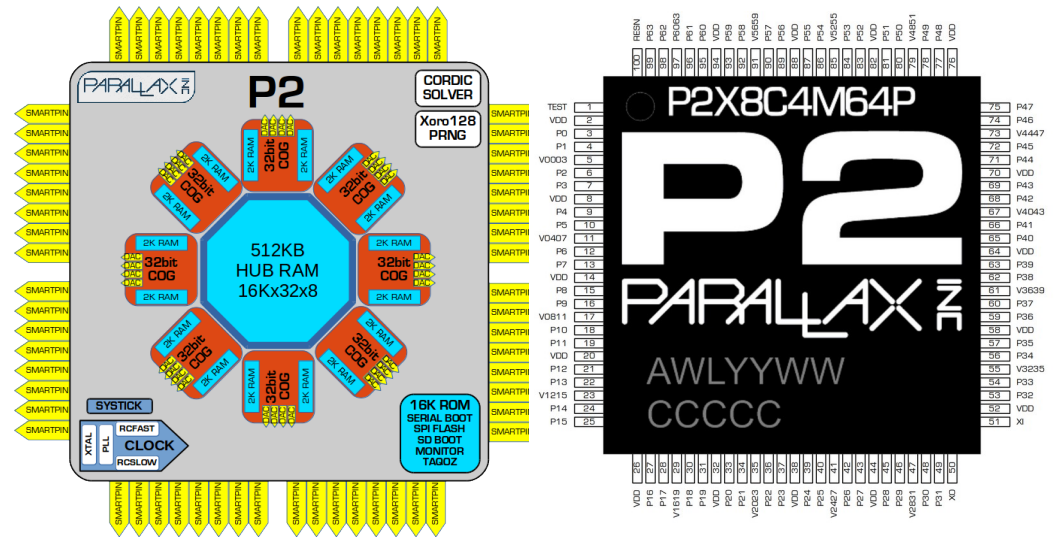
- High-quality PRNG (Xoroshiro128**) updates every clock, unique data per cog and pin

CLOCK - RCSLOW, RCFAST or EXT, XTAL --> PLL = +/-1.64 ; x1..1024 ; +/-2,4,6..30

SMART PIN (64 channels)

- 8-bit, 120-ohm (3ns) and 1k-ohm DACs
16-bit oversampling, noise, and high/low digital modes
Delta-sigma ADC with 5 ranges, 2 sources, and VIO/GIO calibration
Logic, Schmitt, pin-to-pin-comparator, and 8-bit-level-comparator input modes
2/3/5/8-bit-unanimous input filtering with selectable sample rate
Incorporation of inputs from relative pins, -3 to +3
Negative or positive local feedback, with or without clocking
Separate drive modes for high and low output: logic/1.5k/15k/150k/1mA/100uA/10uA/float
Programmable 32-bit clock output, transition output, NCO/duty output
Triangle/sawtooth/SMPS PWM output, 16-bit frame with 16-bit prescaler
Quadrature decoding with 32-bit counter, both position and velocity modes
16 different 32-bit measurements involving one or two signals
USB full-speed and low-speed (via odd/even pin pairs)
Synchronous serial transmit and receive, 1 to 32 bits
Asynchronous serial transmit and receive, 1 to 32 bits, up to clock/3 (60Mbps)

BLOCK DIAGRAM and PINOUT



P2X8C4M64P PIN CONNECTIONS - TQFP100

| PIN | NAME | NOTES | * | PIN | NAME | NOTES | * | PIN | NAME | NOTES | * | PIN | NAME | NOTES | SPI | SD | SER | * |
|-----|-------|----------------|---|-----|-------|-------------|---|-----|-------|----------------------|---|-----|-------------------------|----------|-----|----|-----|-----|
| 1 | TEST | connect to GND | | 26 | VDD | CPU 1.8V | 4 | 51 | XI | Crystal/Ext Clock In | 7 | 76 | VDD | CPU 1.8V | | | | 4 |
| 2 | VDD | CPU 1.8V | 4 | 27 | P16 | SmartPin | | 52 | VDD | CPU 1.8V | 4 | 77 | P48 | SmartPin | | | | |
| 3 | P0 | SmartPin | | 28 | P17 | SmartPin | | 53 | P32 | SmartPin | | 78 | P49 | SmartPin | | | | |
| 4 | P1 | SmartPin | | 29 | V1619 | I/O 3.3V | 5 | 54 | P33 | SmartPin | | 79 | V4851 | I/O 3.3V | | | | 5 |
| 5 | V0003 | I/O 3.3V | 5 | 30 | P18 | SmartPin | | 55 | V3225 | I/O 3.3V | 5 | 80 | P50 | SmartPin | | | | |
| 6 | P2 | SmartPin | | 31 | P19 | SmartPin | | 56 | P34 | SmartPin | | 81 | P51 | SmartPin | | | | |
| 7 | P3 | SmartPin | | 32 | VDD | CPU 1.8V | 4 | 57 | P35 | SmartPin | | 82 | VDD | CPU 1.8V | | | | 4 |
| 8 | VDD | CPU 1.8V | 4 | 33 | P20 | SmartPin | | 58 | VDD | CPU 1.8V | 4 | 83 | P52 | SmartPin | | | | |
| 9 | P4 | SmartPin | | 34 | P21 | SmartPin | | 59 | P36 | SmartPin | | 84 | P53 | SmartPin | | | | |
| 10 | P5 | SmartPin | | 35 | V2023 | I/O 3.3V | 5 | 60 | P37 | SmartPin | | 85 | V5255 | I/O 3.3V | | | | 5 |
| 11 | V0407 | I/O 3.3V | 5 | 36 | P22 | SmartPin | | 61 | V3639 | I/O 3.3V | 5 | 86 | P54 | SmartPin | | | | |
| 12 | P6 | SmartPin | | 37 | P23 | SmartPin | | 62 | P38 | SmartPin | | 87 | P55 | SmartPin | | | | |
| 13 | P7 | SmartPin | | 38 | VDD | CPU 1.8V | 4 | 63 | P39 | SmartPin | | 88 | VDD | CPU 1.8V | | | | 4 |
| 14 | VDD | CPU 1.8V | 4 | 39 | P24 | SmartPin | | 64 | VDD | CPU 1.8V | 4 | 89 | P56 | SmartPin | | | | |
| 15 | P8 | SmartPin | | 40 | P25 | SmartPin | | 65 | P40 | SmartPin | | 90 | P57 | SmartPin | | | | |
| 16 | P9 | SmartPin | | 41 | V2527 | I/O 3.3V | 5 | 66 | P41 | SmartPin | | 91 | V5659 | I/O 3.3V | | | | 5 |
| 17 | V0811 | I/O 3.3V | 5 | 42 | P26 | SmartPin | | 67 | V4043 | I/O 3.3V | 5 | 92 | P58 | SmartPin | DO | DO | | |
| 18 | P10 | SmartPin | | 43 | P27 | SmartPin | | 68 | P42 | SmartPin | | 93 | P59 | SmartPin | DI | DI | | 3 |
| 19 | P11 | SmartPin | | 44 | VDD | CPU 1.8V | 4 | 69 | P43 | SmartPin | | 94 | VDD | CPU 1.8V | | | | 4 |
| 20 | VDD | CPU 1.8V | 4 | 45 | P28 | SmartPin | | 70 | VDD | CPU 1.8V | 4 | 95 | P60 | SmartPin | CK | CS | | 2 |
| 21 | P12 | SmartPin | | 46 | P29 | SmartPin | | 71 | P44 | SmartPin | | 96 | P61 | SmartPin | CS | CK | | 1 |
| 22 | P13 | SmartPin | | 47 | V2831 | I/O 3.3V | 5 | 72 | P45 | SmartPin | | 97 | V6063 | I/O 3.3V | | | | 5 |
| 23 | V1215 | I/O 3.3V | 5 | 48 | P30 | SmartPin | | 73 | V4447 | I/O 3.3V | 5 | 98 | P62 | SmartPin | | | | TXD |
| 24 | P14 | SmartPin | | 49 | P31 | SmartPin | | 74 | P46 | SmartPin | | 99 | P63 | SmartPin | | | | RXD |
| 25 | P15 | SmartPin | | 50 | X0 | Crystal out | | 75 | P47 | SmartPin | | 100 | $\overline{\text{RES}}$ | RESET | | | | |
| | | | | | | | | | | | | PAD | GND | COMMON | | | | |

NOTES:

- *1 – Add pull-up for SPI Flash detection
- *2 – Do not use ext pull-up – uses SD card's internal pull-up for card detection
- *3 – Pull-down disables all serial boot functions including monitor and TAQOZ.
- *3 – Pull-up forces serial boot priority

- *4 – Connect all VDD pins to 1.8V CPU supply
- *5 – Connect all VIO pins to I/O supply voltage
- *6 – 10 to 20MHz crystal input or external clock

P2X8C4M64P TQFP100 LD MECHANICAL

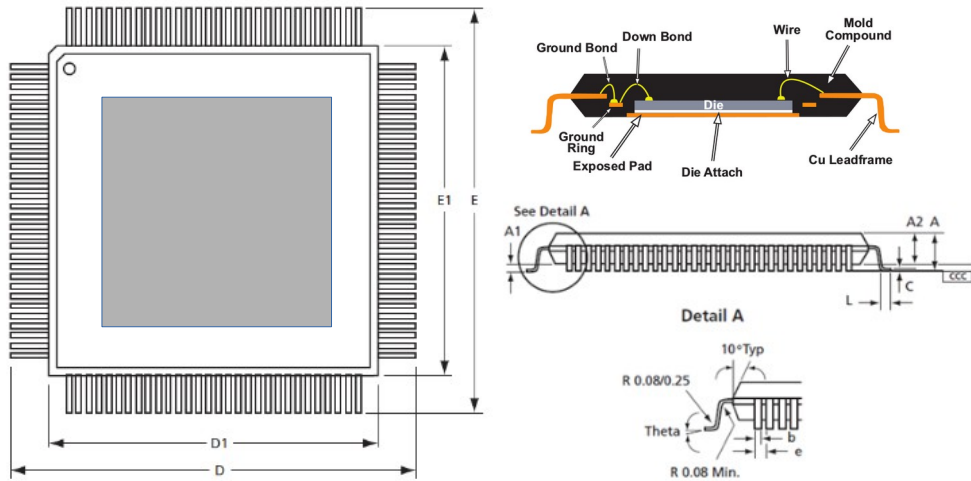


Figure 2: TQFP100 package outline drawings

| Symbol | Description | Min | Typ mm | Max |
|--------|-----------------|------|---------|------|
| D1/E1 | Body | | 14 BSC | |
| D/E | Tip-Tip | | 16 BSC | |
| ePad | Exposed GND Pad | | 10.3 | |
| A2 | Body Thickness | 1.35 | 1.4 | 1.45 |
| A | Height off PCB | - | - | 1.6 |
| A1 | PCB clearance | 0.05 | - | 0.15 |
| e | Lead spacing | | 0.5 BSC | |
| b | Lead Width | 0.17 | 0.22 | 0.27 |
| L | Foot Length | 0.45 | 0.6 | 0.75 |

PHYSICAL PINOUT DIAGRAM



LINKS

- [P2 DOCUMENTATION INSTRUCTION SET & SAMPLES](#)
- [Parallax P2 Forum](#)
- [Spin2gui fastspin IDE](#)
- [TAQOZ](#)
- [SD BOOTER](#)
- [MONITOR](#)

P2 TOOLS

Spin2 GUI with fastspin, p2asm, p2load

The screenshot shows the Spin 2 GUI with a window titled "Spin 2 GUI". The window has a menu bar (File, Edit, Options, Commands, Help) and a toolbar (Compile, Run Binary, Compile & Run). The main text area contains a Spin program named "fibonacci.spin".

```

1  '' simple fibonacci program
2  COM
3  _clkmode = xtall + pll16x
4  _clkfreq = 80_000_000
5
6  OBJ
7  ser: "PrintfSerial"
8
9  PUB demo | i, n, t
10 ser.start(115_200)
11 repeat i from 1 to 9 step 1
12   t := CMT
13   n := fiborec(i)
14   t := CMT - t
15   ser.printf( "fibonacci(%d) = %d; cycles = %d\n", i, n, t )
16
17 '' iterative version
18 PUB fibolp(n) : r | lastr
19   r := 1
20   lastr := 0
21   repeat n-1
22     (lastr,r) := (r,r+lastr)
23

```

Below the code is the "Compiler Output" section, which shows the assembly code for the program:

```

fibonacci.spin
|-PrintfSerial.spin
fibonacci.p2asm
Done.
Program size is 2400 bytes

```

The assembly code is as follows:

```

004cc  02A284F1 | sub    arg1, #2
004d0  C4FFDFFD | calla  #@_Fiborec
004d4  2B6600F6 | mov    Fiborec_tmp008_, result1
004d8  326200F6 | mov    Fiborec_tmp001_, Fiborec_tmp005_
004dc  336200F1 | add    Fiborec_tmp001_, Fiborec_tmp008_
004e0
004e0 | L__0012
004e0 | mov    result1, Fiborec_tmp001_
004e4  5F6704FB | rdlong Fiborec_tmp008_, --ptr
004e8  5F6504FB | rdlong Fiborec_tmp005_, --ptr
004ec  5F6304FB | rdlong Fiborec_tmp001_, --ptr
004f0  5F8304FB | rdlong _Fiborec_N, --ptr
004f4
004f4 | _Fiborec_ret
004f4  2E0064FD | reta

```

ROM TOOLS - TAQOZ

```

-----
Parallax P2  ...--TAQOZ--...  V1.0--142  180530-0135
-----
TAQOZ# WORDS
DUP OVER SWAP ROT -ROT DROP 3RD 4TH 2DROP 3DROP NIP 2SWAP 2DUP ?DUP AND
ANDN OR XOR ROL ROR >> << SAR 2/ 2* 4/ 4* 8<< 16>> 8>> 9<< 9>> REV |< >|
>N >B >9 BITS NOT = <> 0= 0<> 0< < U< > U> <= => WITHIN DUPC@ C@ W@ @ C+!
C! C@++ W+! W! +! ! BIT! SET CLR SET? 1+ 1- 2+ 2- 4+ + - UM* * W* / U/
U// // */ UM// C++ C-- W++ W-- ++ -- RND GETRND SQRT SETDACS ~ ~ W~ W~~
C~ C~~ L>S >W L>W W>B W>L B>W B>L MINS MAXS MIN MAX ABS -NEGATE ?NEGATE
NEGATE ON TRUE -1 FALSE OFF GOTO IF ELSE THEN BEGIN UNTIL AGAIN WHILE REPEAT
SWITCH CASE@ CASE= CASE> BREAK CASE ADO DO LOOP +LOOP FOR NEXT ?NEXT I
J LEAVE IC@ I+ BOUNDS H L T F R HIGH LOW FLOAT PIN@ WRPIN WXPIN WYPIN RDPIN
RQPIN AKPIN WAITPIN WRACK PIN @PIN ns PW PULSE PULSES HILO DUTY NCO HZ
KHZ MHZ MUTE BLINK PWM SAW BIT BAUD TXD RXD TXDAT WAITX WAITCNT REBOOT
RESET 0EXIT EXIT NOP CALL JUMP >R R> >L L> !SP DEPTH COG@ COG! LUT@ LUT!
COGID COGINIT COGSTOP NEWCOG COGATN POLLATN SETEDG POLLEDG KEY WKEY KEY!
CON NONE COM CONKEY CONEMIT SEROUT EMIT EMITS CRLF CR CLS SPACE SPACES
RAM DUMP: DUMP DUMPM DUMPL DUMPA DUMPAW QD QW DEBUG lso COG LUT KB MB
M . PRINT .AS .AS" .DECL .DEC4 HOLD #> <# # #S <D> U. .DEC .BIN .H .B .BYTE
.W .WORD .L .LONG .ADDR PRINT$ LEN$ " ." CTYPE ?EXIT DATA? ERASE FILL CMOVE
<MOVE s ms us CNT@ LAP LAP@ .LAP .ms HEX DEC BIN .S WORDS @WORDS GET$
SEARCH $># @DATA HERE @HERE @CODES uemit ukey char delim names TASK REG
@WORD SPIN | | , [W] ["] NULL$ $! $= ASM FORGET CREAT$ CREATE VAR pub
pri pre ; ; [ ] ' := ==! ALIGN DATCON ALLOT org bytes words longs byte
word long res [C] GRAB NFA' CPA CFA \ --- ( { } IFNDEF IFDEF TAQOZ TERM
AUTO SPIRD SPIRDL SPIWB SPICE SPIWC SPIWW SPIWM SPIWL SPIPINS SPIRX SPITXE
SPITX WSLED WAIT CLKDIV RCSLOW HUBSET WP WE CLKHZ ERROR SFPINS SF? SFWE
SFINS SFWD SFSID SFJID SFER4 SFER32 SFER64 SFERASE SFWRPG BACKUP RESTORE
SFRDS SFWRS SFC@ SFW@ SF@ SF .SF SDBUF sdpins MOUNT DIR !SD !SX SD? CMD
ACMD cid SDWR SDRDS SDWRS FLUSH FOPEN FLOAD FGET FREAD FWRITE SECTOR SDRD
SDRDS SDADR SD@ SD! SDC@ SDC! SDW@ SD @FAT @BOOT @ROOT fat END 433 ok

```

SD-BOOTER & MONITOR