Digital IO on Green Arrays' GA144 implemented in polyFORTH®

Peter Milford,

Outline

- SPI introduction
- CMA3000 3 axis accelerometer
- Implementation
  - Setting up and using paths.
  - Toggling IO pins.
- Higher level application: Fall detection

SPI protocol summary

- SPI is a synchronous serial protocol.
- SPI uses:
  - Master clock
  - Master out slave in data line (MOSI)
  - Master in slave out data line (MISO)
  - Chip select
- The length of transfers, sense of the clock, and phasing can change.

CMA3000 timing

CMA3000 SPI frame format and transfer protocol is presented in Figure 7.

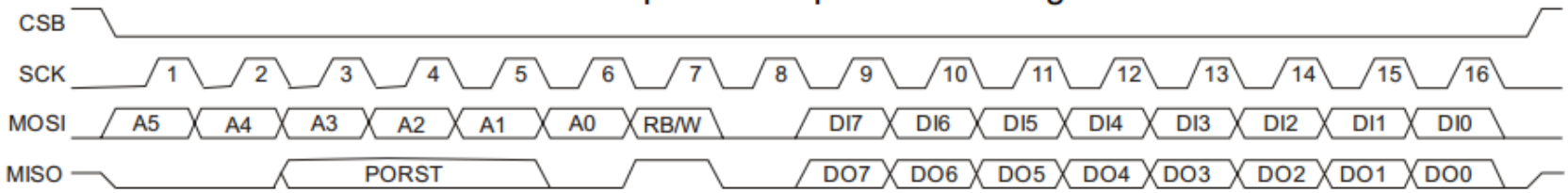


Figure 7. SPI frame format

Each communication frame contains 16 bits. The first 8 bits in MOSI line contains info about the register address being accessed and the operation (read/write). The first 6 bits define the 6 bit address for the selected operation, which is defined by bit 7 (‘0’ = read ‘1’ = write), which is

Change output bit in IO register

```
0x1DA CONSTANT IO
ND N517 3 uu 8 rr 0 path,  ( Define path to 517)
: HIGH ( -- )   0 3  IO R! ;
: LOW  ( -- )   0 0  IO R! ;

N517  ( Select path to node 517)
HIGH  ( Set IO pin to high)
LOW   ( Set IO pin to low)
```

SPI Low level words

- Lines to control/access: CS, CLK, MISO, MOSI
- Chip select. Active Low. As we are toggling a number of lines, re-select path on each access

```
ND CS  4 rr  1 dd  4 rr  path,   ( On analog out port 117)
: CSH  ( --)  CS  CSR 3  IO R! ;   ( CSR 1FF xor 0x155)
: CSL  ( --)   CS  CSA 0  IO R! ;   ( CSA 0  xor 0x155)
ND CLK  8 rr  0  0  path,  ( N217)
: CLKH  ( --)  CLK  0 3  IO R! ;
: CLKL  ( --)   CLK  0 0  IO R! ;
```

SPI Control (Cont)

```
ND MOSI   2 uu  8 rr  0  path, ( N417)
( Output a low order bit.)
: MOSISET  (bit --) MOSI  1 AND 2 OR 0 SWAP  IO R! ;

ND MISO  3 uu  8 rr  0  path, ( N517)
( Return value of IO pin)
: MISOGET (-- bit)  MISO  IO R@ SWAP DROP 2/ ;
```

Oneword SPI transfer

- Simultaneous output a word high order bits first and input a word, high order bits first.

```
: MSB (n – signn) –IF 1 ELSE 0 THEN ;

: ONEWORD ( out—in)  CSL  15     FOR
    DUP  MSB MOSISET CLKH
    MISOGET CLKL SWAP 2*  OR     NEXT CSH ;
```

CMA3000 access words

```
( Register numbers in CMA3000)
5 CONSTANT XA   6 CONSTANT YA   7 CONSTANT ZA
( Build register access command structure)
: (REG) ( d REG r/w– dat)  SWAP 2* 2* OR 1 OR ONEWORD ;
: REG!   ( dat REG --) 1 (REG) DROP ;
: REG@ ( REG–dat)  0 (REG) ;
( Convert 8 bit signed to 16 bit signed during read)
: SEXT ( S8 – S16)  ><   2/ 2/ 2/ 2/ 2/ 2/ 2/ 2/ ;
: ACEL@  REG@  255  AND SEXT ;

: XYZ  ( -- Za Ya Xa) ZA ACEL@  YA ACEL@  XA ACEL@ ;
```

Fall detection

- Detect a fall by total acceleration about 0

```
: TACCEL  (Ax Ay Az --- A^2)
   DUP *   ROT
   DUP *   ROT
   DUP *   + + ;

(  Exit when unit falls)
: FELL?  BEGIN XYZ TACCEL 200 <  END ;
```

Questions?

Contact:

Peter Milford,

pmilford@parallel-rules.com

Or

GreenArrays.

References

- GA144/F18A
  http://www.greenarraychips.com/home/documents/greg/DB001-110412-F18A.pdf
  http://www.greenarraychips.com/home/documents/greg/PB004-110412-F18A-IO.pdf
  http://www.greenarraychips.com/home/documents/greg/DB002-110705-G144A12.pdf
- CMA3000 data sheets:
  http://www.vti.fi/sites/default/files/documents/cma3000_d01_datasheet_8277800a.03.pdf
  http://www.vti.fi/sites/default/files/documents/cma3000-d0x_product_family_specification_8281000a.04.pdf
- SPI protocol
  http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

Backup Slides:

GA144

- Green Arrays 144 processor chip.
- Arranged as 8 x 18 processors.
- IO pins attached on edge processor nodes only. Some IO is special purpose.

F18A block diagram

Here is an introduction to the resources within each F18A computer. All elements are 18 bits wide unless otherwise noted.

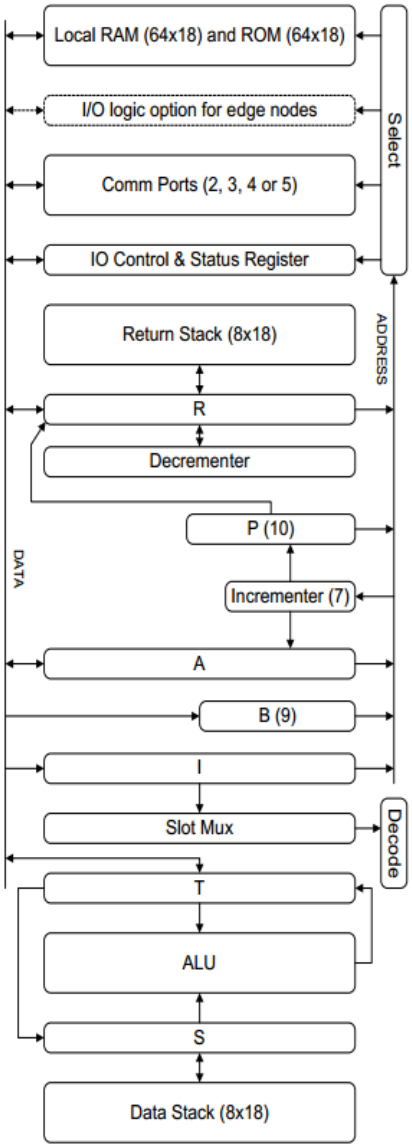
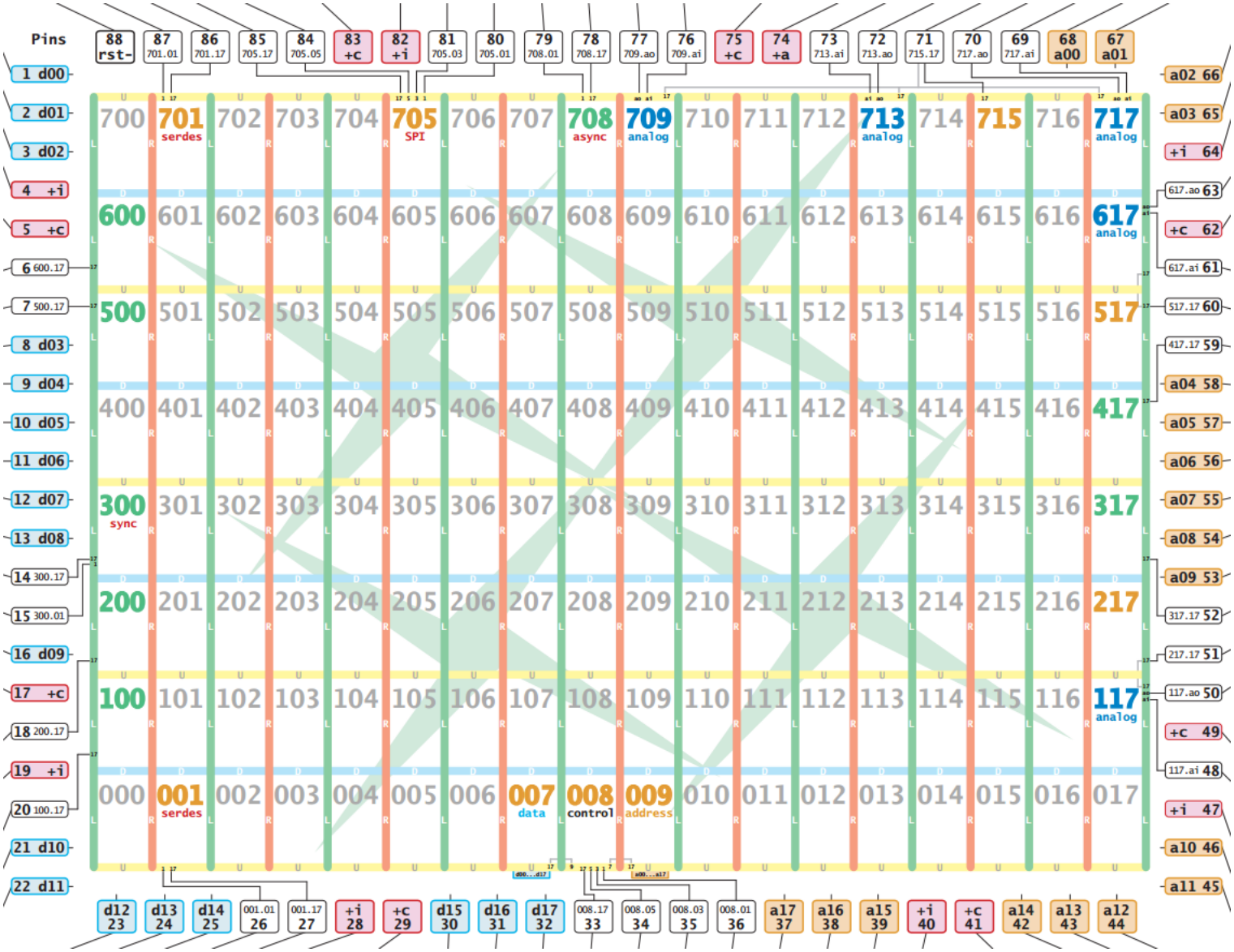


Figure 1 F18A Block Diagram

- **RAM**: 18 bits wide, 64 words in each node.
- **ROM**: 18 bits wide, 64 words in each node.
- **I/O logic** may be present on any edge node of a chip.
- **Communication Ports** are configured based on position within the array (four for inner; three ports array edges; two for corners.) One additional port is optional.
- **IO** control and status for communication ports and I/O logic.
- **Return Stack** consists of 8 registers addressed circularly (no fixed top or bottom.) Pushing a word onto this stack makes it the new "top" item, overwriting the former "bottom" item. After popping 8 items, those 8 items are repeated.
- **R** extends this stack to 9 entries but only 8 are circular. When used as a loop counter, **R** decrements toward zero.
- **P** serves as a "program counter", holding the address of the next word in the instruction stream. Within RAM or ROM, it is incremented after each word is fetched, circularly within whichever storage class it currently points to. When **P** addresses I/O space, it is not incremented. **Bit P9** enables Extended Arithmetic Mode when set.
- **A** is a general purpose read/write address or data register.
- **B** is a write only address register, containing on reset the address of **io** register.
- **I** is the instruction register into which instruction words containing 1, 2, 3 or 4 opcodes are fetched for execution.
- **T** is the top word of a 10-element data stack.
- **The ALU** (Arithmetic/Logic unit) performs unary operations on **T**, binary operations on **S** and **T**, or a multiply step involving **S**, **T**, and **A** as directed by opcodes. A carry latch is involved in some Extended Arithmetic mode instructions.
- **S** is the second word of a 10-element data stack.
- **Data Stack** is 8 circularly addressed registers constituting the third through 10th elements of a 10-element stack.

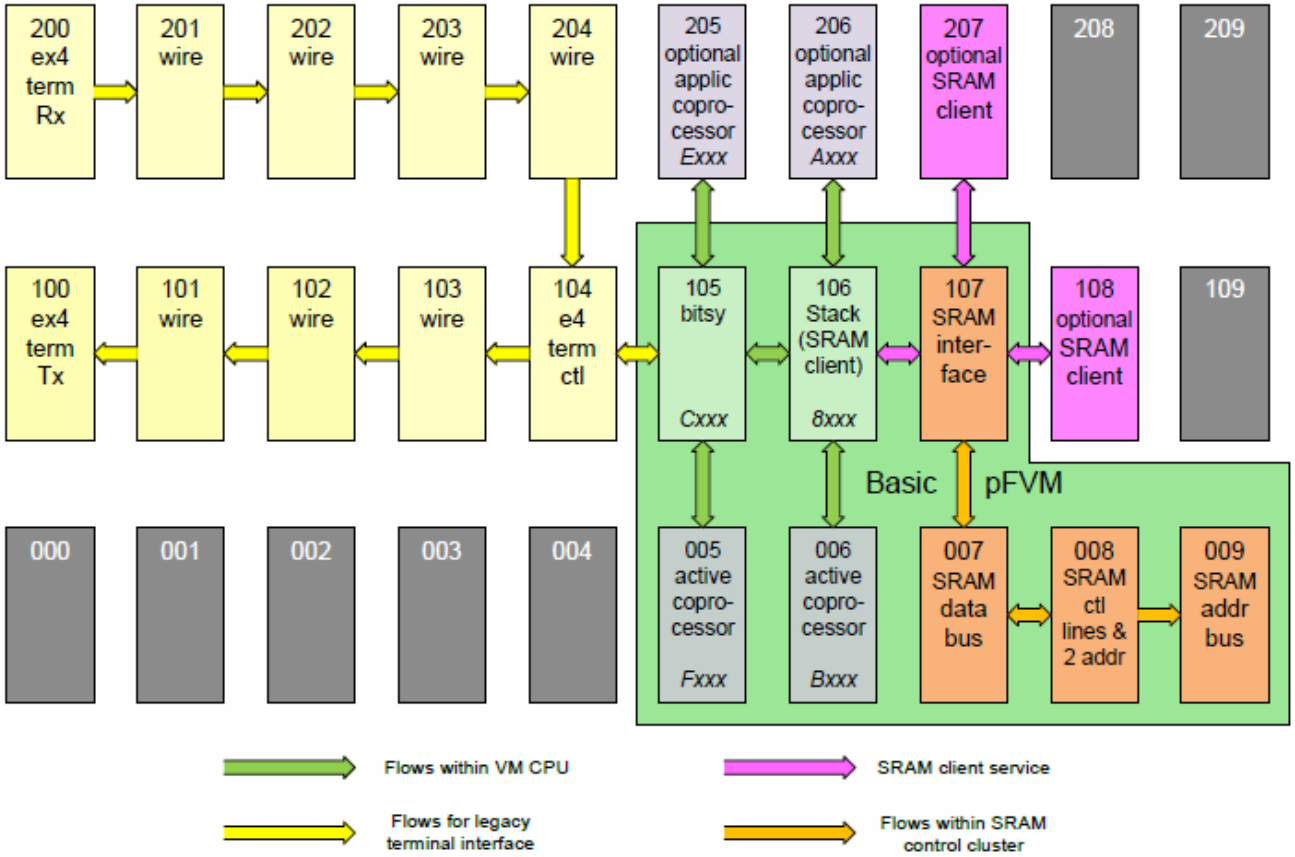
GA144 block diagram



polyFORTH Virtual Machine

Based on eFORTH VM by Bill Muench and John Rible

The polyFORTH Virtual Machine (pFVM) is implemented by a virtual CPU consisting of six nodes (005, 006, 105, 106, 205 and 206.) This CPU is one of the three principal clients of the external SRAM control cluster, consisting of four nodes (107, 007, 008 and 009.) In the present version, nodes 205 and 206 are entirely available for application microcode and the basic CPU is implemented by the remaining four nodes. The eight nodes constituting the basic pFVM are shown highlighted with green background in this diagram of their immediate neighborhood:



polyFORTH Virtual Machine for GA144-1.2

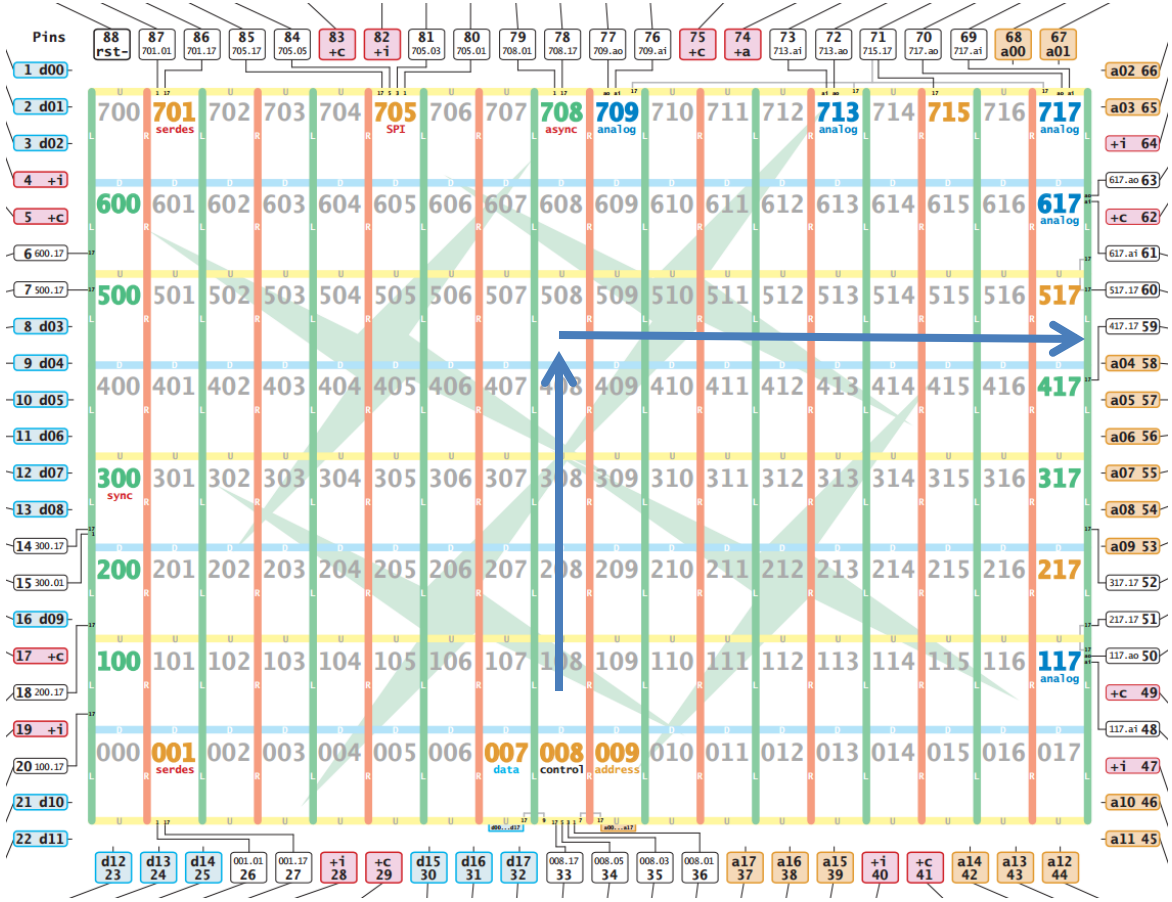
Paths to nodes.

- High level words Implemented in polyFORTH.
- **path,** word takes 3 pairs of numbers on stack and builds an 18 bit ‘path’ to a destination node.
- For convenience, direction constants **rr**, **uu**, **ll**, **dd** are defined for right, up, left and down.
- **ND** is a defining word that takes name and path and creates a word to implement the path.

R! R@ remote node memory access words

- Once a path has been setup by executing a word to define a path (for example **N517**)
- The remote node's memory can be accessed:
  - **R@ (a—n)** fetch word on current path at given address
  - **R! (d a--)** store (double) word at given address on current path.
  - The path stays setup till changed.

Sample path to node



ND N517 3 uu 8 rr 0 0 path,