Military Technical College, Kobry El-Kobbah, Cairo, Egypt

9th International Conference On Aerospace Sciences & Aviation Technology

VLSI DESIGN OF DIGITAL CORRELATORS FOR GPS RECEIVERS*

Bahnas M., Al-Imam M.***, Ragaie H. F.******

ABSTRACT

There are an increasing number of applications requiring precise relative position and clock-offset information. The Global Positioning System has demonstrated precise and drift free position and timing information using Code-Division-Multiple-Access (CDMA) spread spectrum technology. From the most interesting applications are the navigation and guidance, they are severely needed in the aviation field. Detecting this accurately with no aid or dependence on anybody is an achieving step towards increasing the safety factors in the plane flights besides many other applications.

This paper explores the GPS system as a whole, then targets to a certain objective. The main objective is to make a VLSI design and implementation for the main digital baseband block in the GPS receiver, the Digital Correlator. We designed 3 architectures, which are the serial, parallel and hybrid Correlators. Each of the serial and the parallel has pros & cons, which are discussed in details. There is a trade-off between the preferred small chip area and the quick response of the GPS receiver to its related stimulus. We finally reached to an optimized design gathering some advantages of both of them, the hybrid correlator. Using Mentor Graphics tools, the design is implemented in ASIC layout and FPGA chip.

KEYWORDS

Global Positioning system, VLSI, Guidance, Navigation, Aviation Electrical systems, GPS Receiver, Digital Correlator, ASIC, and FPGA.

---

* A graduation project work (yr.1999/2000), Ain Shams University, Cairo, Egypt.
** IC Design Consultant, Mentor Graphics Egypt, Cairo, Egypt.
*** Graduate, Electronics & Communications Engineering Dept., Ain Shams Univ., Egypt.
**** Professor, Electronics & Communications Engineering Dept., Ain Shams Univ., Egypt.

I. INTRODUCTION

In the Global Positioning System (GPS), a receiver determines its physical position based on the signals transmitted from the 24 orbiting GPS satellites. Each satellite transmits direct-sequence spread-spectrum precise (P) code and coarse/acquisition (C/A) code in orthogonal phases. The P-code was highly restricted and was available only for military uses, whereas the C/A code is available for commercial applications. Recently, the US government opened the (P) code for public usage. However, our GPS receiver is designed to receive the C/A code only. The C/A code is a 1023-bit, repeating 1.023 MHz pseudorandom noise (PN) code. Each satellite has a unique C/A code and broadcasts a 1500-bit block of navigation data at 50 b/s [4,5].

A GPS receiver determines its position by first estimating its relative distance, or pseudorange, to at least four GPS satellites. Knowing these satellites' locations and the corresponding pseudorange estimates, the receiver can determine its position by solving a set of equations with four unknowns, which are the three spatial dimensions of (x, y, z) and the clock offset between the local and the satellites oscillators. Satellite coordinates are transmitted within its unique 50 Hz navigation data. To obtain these pseudo-range estimates, the receiver must synchronize with each of the satellites transmitted spread-spectrum signal.

The GPS receiver is employed in a wide variety of applications mainly in the Navigation and Guidance fields. It was originally launched for military purposes such as directions and ways detector for vehicles and foot soldiers, missiles director and minesweepers. Concerning the Aviation field, it provides performance compliant with the stringent requirements for aircraft precision approach and landing. It is applied to inform the pilot with the plane current position so he can relate the data with other targeted locations. It can also be used as a tracking tool for the path of the flying trip. For GPS receivers to be employed in such mobile electronic devices, the receiver must be highly integrated and power efficient. Integration reduces cost and size, and low power extends the battery life of the unit.

Our designs acquired specifications matching the pre-mentioned applications. It is somehow a conventional digital receiver, so as to get use of advantages of digital over analog implementation in a large portion of the chip. Our work here is concerned with the design of the major digital block in the receiver; the correlator. The digital correlator, with the aid of the processor, is the brain of the receiver in the scope of managing the data extraction from the visible GPS satellites in the sky. We went through "three" architectures for the correlator to reach for more optimized solution concerning the trade-off between the applications requirements and the VLSI implementation.

Those architectures are:

- The serial correlator
- The parallel correlator
- The hybrid correlator

Each one of them possesses certain advantages over the others; the main two optimization extremes are minimizing either the chip area or the satellites tracking response time.

II. GPS RECEIVER ARCHITECTURE

Digital Receiver

The majority of early GPS receiver designs made extensive use of analog signal processing techniques, however, most modern receivers incorporate digital signal processing wherever possible. Our GPS receiver uses a mostly digital architecture. Analog signal processing is limited to preselection and gain applied to the GPS signals during down conversion with fixed translation frequencies. In a digital receiver, analog to digital (A/D) conversion takes place at the receiver IF. Code correlation and further signal processing occur digitally. The illustrating block diagram is as shown in fig.1.

GPS Signal Components

The signal at a GPS satellite antenna is a combination of the three components: carrier wave, ranging codes and navigation message. The generation of the signal to be transmitted is carried out in a number of steps: First step is binary-to-binary conversion. Because of the difference in frequency between the C/A code and the message stream this has the effect of inverting 20 C/A code binary "states" whenever the data bit of the navigation message is equal to "1". Conversely, when the data bit is "0" the C/A code sequence remains unaffected. The same satellite message is also modulated onto the P code sequence using modulo-2 addition procedure. Second step is BPSK modulation to two L-band carriers with the previous two composite signals (fig.2).

C/A-Code

The C/A-code consists of a 1023 bit pseudorandom noise (PRN) code with a clock rate of 1.023 MHz which repeats every 1 msec. The short length of the C/A-code sequence is designed to enable a receiver to rapidly acquire the satellite signals. A unique PRN is assigned to each GPS satellite and selected from a set of codes called *Gold codes*. The Gold codes are designed to minimize the probability that a receiver will mistake one code for another (minimum cross-correlation). Both the C/A and P code generating algorithms are known, and are based on a simple tapped feedback shift register scheme. The logic implementation of the C/A code generator is shown in fig.3.

GPS Receiver Operation

In order for the GPS receiver to calculate a PVT (Position Velocity Time) solution, it must fulfill these consecutive operations:

- Down conversion of the RF signal.
- A/D conversion and Demodulation.
- Digital signal processing.
- Search for a PRN C/A code lock.
- Obtain bit synchronization with the navigation message, and then frame synchronization.
- Take range and range rate measurements.
- Search, acquire and track 2nd to 4th satellites.
- Solve for range equations.

III. VLSI DESIGN OF CORRELATORS

The detailed role of the digital correlator within the GPS receiver is illustrated in fig.4. It can be implemented in one of the following architectures.

A. Serial Correlator

Serial correlator or sliding correlator achieves correlation between received signal from the satellite and the signal generated in the receiver in the following steps:

1. Serially modulo-two adding each received bit with a generated bit.
2. A counter counts the output of the XOR gate over a complete epoch.
3. At the end of the epoch the result is compared to a threshold value to decide whether the signals are in correlation or not, if in correlation it decides the logic value of the received data.
4. If no correlation the generated code is delayed one chip and the process is repeated as above.
5. If the generated code is delayed a number of times equal to the number of bits in one epoch and no correlation is found, the correlator knows that this code is not present in the received codes and switches to anther code to check for it.
6. The above step is repeated until the receiver finds a match (it is *locked* to a satellite).

Serial correlator block diagram is in fig.5.

- *CA/Code Generator:* generates the PRN codes sent by the satellites.
- *Xor gate:* for modulo two addition between received code and the generated code.
- *Counter:* is resetted at the beginning of each epoch and then counts the output of the xor gate.
- *Latch:* passes the output of the counter to the threshold circuit at the end of the epoch.
- *Threshold circuit:* contains predefined margins to which the counter output is compared. Values of these margins are thoroughly examined in a next section. It gives 2 outputs:
    - *Flag:* gives an indication of the state of the correlator, i.e in lock state or still searching for the code.
    - *Data:* when the correlator locks with a code this pin states the logic level imposed on the found code.
- *Clock control:* controls the clock of the C/A generator and has two modes:
    - C/A generator misses a clock event at the beginning of each epoch in such a way the generated code is delayed on bit with respect to the received code.
    - Free run when the correlator is in lock state.

The logic design of the clock control block is shown in fig.6.

Simulation results waveforms are shown in fig.7.
clk is the clock signal from the clock generator.
ck is the signal that triggers the CA/Code generator as seen, a ck event is missed every epoch. This is achieved by masking a one clk event at the beginning of the epoch with the aid of the T flip-flop and a mux.

Advantages of the serial correlator are:

- Its small size to consume small silicon area on chip.
- Once in lock state, extraction of data imposed on PRN code is performed in a satisfactory time.

Disadvantage of the serial correlator that;

- It takes long time to lock. A worst case is = "number of chips per epoch x epoch duration" which is equal to 1023 msec.

B. Parallel Correlator

The parallel correlator design is directed to overcome the disadvantage of the serial Correlator. The concept is that we need to compare 1023 bits stream to another 1023 bits stream, bit to bit, 1023 comparison simultaneously, then turn to a new 1023 bits comparison and so on. To detect which satellites are visible now in the sky, the correlator correlates all the 32 codes with the received bit stream, and then decides which ones are available. This is done one satellite by one consecutively. Choosing the code as the output of the generator and when to switch to another code is operated by the mux 32-to-1 block. To compare 2 bit streams the length of each is 1023, we need a pair of 1023 bit shift registers. Each corresponding 2 bits in both shift registers are modulo-2 added together. The output of the 1023 xor gate is the input of the 1023 bit adder. The block diagram of the parallel correlator is shown in fig.8.

We consider both code versions have the same clock due to the action of the PLL implemented in the receiver. This is done by adjusting the clock of the code generator to follow the received code rate. The detection operation accommodates 2 functional modes consequently to fulfill its role. The first is the *sliding mode*: the shift register connected to the code generator is filled with the 1023 bit stream (by enabling the clock continuity to its clk input). During this the received data filling the other shift register is ignored leaving the clock enabled all the time. The second mode is the *stationary mode*: now assuming that the lower shift register (filled with the generated version) has been clock disabled so that the data held in each D-flip flop is fixed (i.e. no sliding). Then without disabling of the clock of the upper shift register, It looks like two plates one is stationary and one is sliding over it. An ideal case where we assume that we begin powering on the receiver at the same instant we receive a code that is completely synchronized to the former with no bit shift between them. The probability of no shift is as low as 1/1023. The practical solution is to try the 1023 probability cases. This is fulfilled by sliding one bit stream and making the other fixed. Sliding rate is 1.023 MHz.

The 1023 bit adder role is to count the number of 1's at the output of the previously mentioned 1023 xor gates array. It must execute this in less than 977.5 nsec (i.e. less than the sliding rate), this is to be able to calculate the value of the 1023 sliding cases. In response we can now detect the shift between beginning of epochs of 2 versions (giving minimum value of 1's) by a1023 decimal counter. Also as an advanced step we can use this feature to synchronize them together (i.e. code tracking).

The architecture of the *1023 bit adder* is illustrated in fig.8. We've reached an optimized design to economize its silicon area. The main idea is that we use an additional clock which is 10 times the sliding rate, this is the substitution for selecting

the architecture with minimum hardware. The concept we adopted is to divide the 1023 bit in 8 subsets, each of 128 bit. Every 8 bit (of the 1023 bit) are entered on the input of an 8-to-1 mux. The clock of this mux is 10.23 MHz. There are 128 mux of this type, so the total output is 128 bit. These bits are the input of a multi-level parallel adder, a detailed description is given in fig.9. This parallel adder consists of 6 hierarchical levels of individual adders. The first level consists of an array of 1 bit adders, the second step consists of an array of 2 bit adders and so on. The last level is one 6 bit adder and its output is 6 bit binary word. Our original task is to add 1023 bit not 128, so now we want to add the 6 bit binary output to another 6 bit binary output from previous additions and repeating 8 times to get total sum of all the 1023 bit vector. The solution is that the output of this adder is entered on an accumulator consisting of a 10 bit adder to add this bit vector to a vector of zeros. This initialization vector of zeros is loaded in this place at every positive edge trigger of the 1.023 MHz. The next step is to switch the 8-to-1 mux's to let the output of each is the second bit. Now, we'll add new 128 bits in the 6 steps adder then the output is added by the 10 bit adder to the value of the previous 128 bits addition stored in the accumulator. The result is stored in the accumulator replacing the old value. The mux's then switch to the 3^{rd} bits, 4^{th} bits till the 8^{th} bits and every time the output of 6 step adder is added to the value stored in accumulator. The final value is stored at the output of the 10 bit adder in a synchronous buffer. The threshold is not allowed to take its decision before the end of the 1.023 MHz clock pulse.

The following block is the decision-maker or the threshold detector. The role of this signal output (named "comp") is the most important in the correlator as a whole. The "comp" goes high at the instant of no shift between the 2 codes and that is the *synchronization* event. The situation now is that the lower shift register is *stationary* and the upper is *sliding*. If the comp output gets high we allow the lower shift register to slide also, both codes will slide in a synchronized mode and will never go out of that state. That is synchronization (tracking).

This design has great advantages over the serial correlator; the main one is the very short time for locking. Assume that the generated code is a subset of the group of the visible satellites in the sky, so worst case of locking is 2msec of which 1msec is for filling the lower shift register with a complete epoch of the code which is constant for all cases This can be the minimum locking time if there is no shift between the two code versions. The maximum shift is 1023 bit which take 1 msec to find it (the concept of sliding). Compared to serial correlator, the later needs 1msec if zero shift and needs 1023 msec if maximum shift. Also, time taken to fetch for all the 32 codes is 32 x 2msec = 64 msec. The serial needs 1023msec x 32 = 32736 msec = 545.6 sec. This high locking speed is needed at high dynamic vehicles on ground or at sea or in sky. This is because calculating velocity depends on rate of change of position measurement not absolute measurements.

The main disadvantage of this architecture is the large layout area, mainly for the 2046 D-flip flops and the 1023 bit adder. This large area can cause high power consumption and so not efficiently integrated into navigation systems. Looking from another point of view, we found that utilization of these shift registers is not all the time since after locking the only job of that correlator is to check that the receiver is still locked to a satellite. This job can be fulfilled by only 1 bit shift register, 1 xor gate

and a serial adder. As interpretation, one can say that system efficiency of this large area design is very low.

Concerning the simulation results of the Parallel Correlator, we entered a certain data at the correlator input from additional code generators representing the transmitters. It is a composite data of several codes to simulate the situation of having several codes in the sky. We made a time shift between them and our correlator code generator. The job of the correlator is to track the codes and find their identity numbers. Illustrative waveforms are shown in fig.10 and fig.11.

C. Hybrid Parallel & Serial Correlator

The serial and parallel correlators are common implementations where each one is suitable for certain requirements. For the GPS, some applications prefer small area than having a very high speed locking so the serial is suitable for it. Other applications will focus on high speed rather than small area. Most GPS receiver manufacturers implement about 8 to 12 channels in the same receiver chip. If they decided to implement the parallel correlator they will face the problem of its large area and high cost. To make such a multi channel correlator, we have to compromise between the two architectures.

We've designed and implemented a new correlator called the hybrid correlator. It is a hybrid design that is formed of one optimized multi-level parallel adder and several serial adders. All are integrated and synchronized together. This design not only still possesses the *high speed locking* of the parallel correlator, but also *increased the system efficiency* to a great extent. Another advantage of this correlator that it can make a multi channel correlator by using *number of serial correlators* equal to the number of channels. The block diagram of a hybrid correlator is shown in fig.12.

The technique of this design has modified only few things in the design of both the serial and parallel designs which are integrated here together. At powering on the receiver, the serial adders are disabled. The job of the parallel correlator now is to fetch and track for codes in the sky. The new behavior appears at the moment when lock to one satellite. The past action of the parallel correlator was to continue locking with that code. Here, we'll let it do a new action: the parallel correlator chooses one serial correlator to enable its clock and orders it to generate the code of this specified satellite and correlate it with the received data. From now on, this serial correlator will lock with that satellite independently from the parallel. As long as this serial correlator is locked to the satellite, its output "flag" will be high. Once not locked (i.e. invisibility) the threshold enforces the "flag" to be low. To choose one serial correlator, the parallel correlator scans on the "flags" output of all the serial correlators and choose anyone whose flag is'0'. It leaves those of flag ='1' since they are already locked to another satellites. Once the parallel correlator chooses a serial one, it maps the same 5 bit vector that the parallel 32-to-1 mux selector holds (i.e. identity of the tracked code) to the serial 32-to-1 mux selector. Now, the output of the serial mux is a replica of the specified code. This can be named *handling and mapping* process made by the parallel correlator to the serial correlator. After fulfilling this handling process, the parallel correlator instead of keeping synchronization with this code, it switches to the next satellite code. The lower shift register will now enter its sliding mode to allow the master code generator (of the

parallel correlator) to fill the shift register with a replica of the new specified code by the mux. and so on.

This last change of attitude of parallel correlator increases its system efficiency to a large extent. This is because it didn't waste time in keeping synchronization with tracked codes and left this easy job to the serial correlators.

Concerning the simulation results for the Hybrid correlator, very similar results to that of the pure parallel correlator are found. We'll focus here only on the main differences between them, especially at the instant of occurrence of correlation. Illustrative waveforms are shown in fig.13 & fig.14.

D. ASIC Layout Design Flow:

The VLSI implementation of the GPS Digital Correlators followed the top-down design methodology. The used design entry is the VHDL code written in a hierarchical style coping with the illustrated blocks partitioning of the system. The used design software tools in the whole design were Mentor Graphics tools:

- Renoir and ModelSim for the VHDL entry and simulation.
- Leonardo for synthesis.
- Design Architect for schematic entry.
- IC Station for place and route of ASIC layout. Also for IC verification steps like DRC (Design Rule Check), LVS (Layout vs. Schematic) and PEX (Parasitic Extraction).

We passed through the whole design flow till reaching an ASIC layout for the hybrid correlator with one parallel correlator & 4 serial correlators, i.e. 4-channel parallel GPS correlator. We used CMOS 0.8-micron AMS technology. The total chip area is 10.44 mm^2. The layout is shown in fig.15.

IV. THRESHOLD MARGINS MEASUREMENTS

As indicated in the previous section the output of the adder (counter) is compared to predefined margins called the threshold margins. The chart describing these margins is in fig.16. Using Matlab calculations, VHDL simulation and hardware measurements implemented on a Xilinix Spartan FPGA chip, we can extract the values of the 2 margins.

Assume that only one code is received by the antenna of the GPS receiver, the code is correlated with itself in the receiver. We simulated this situation using Matlab. A spike is obtained at the instance where the codes are in phase otherwise the value of the counter is in the noise range. The difference in level between the spike and the noise is relatively very high. Depending on the navigation message bit logic level; the spike is either over or below the noise level. If logic zero is imposed, the response curve helps us to determine the lower threshold margin. If logic one is imposed, the response curve helps us to determine the upper threshold margin, (fig.17).

Fortunately these results are the same for all other codes if treated solely. However, this is not the situation in practice, at least four codes are present together so we must see the impact of presence of many codes on the threshold margins. The

difference in level between the spike and the noise is relatively less than the previous ideal case, this is depicted in fig.18.

Knowing that there are superposition of codes since they are sent using BPSK modulation. The RF signals from all visible satellites enter the receiver at the same time and seen by the front end as one superimposed signal. The expected behavior of the superposition of the simultaneous bits of different codes is logically modeled [1], in order to calculate the real threshold margins.

We implemented a circuit to calculate the value of correlation between the superposition signal and the generated code at every possible shift and outputs the result to a 10 bit binary word to a digital to analog converter "DAC" where the analog output signal is viewed on a digital CRO. The block diagram of the circuit is shown in fig.19.

The result is captured and displayed in fig.20. Noticing that the pattern is repeated every epoch shift time, we used a 50 MHz clk frequency that gives "50,000,000 / 1023" frequency pattern which is equal to 48 kHz.

V. CONCLUSION

The architecture and implementation of the digital correlators for GPS receivers are presented. We went through 3 designs for the correlator. The serial correlator is preferred because of its small chip consuming area but not preferred in its response time of locking to GPS satellites transmitted codes. The parallel correlator improved this response time to a great extent, but on the other hand its chip area is excessively large. Besides that this large area is not on duty all the time so time-inefficient solution. The hybrid correlator is an improved version of both previous correlators integrated together in one system. This design has the advantage of the quick response of the parallel correlator besides increasing the system efficiency by keeping it on duty all the time. This last property was acquired by assigning the easy jobs (routine check on continuity of locking), previously done by the parallel correlator, to several serial correlators so that the parallel one is now dedicated to the main jobs (searching for codes to lock to) affecting the time response. We implemented two designs, one is ASIC layout for the hybrid correlator using CMOS 0.8 um technology. The other is a FPGA chip for the serial correlator to do some hardware measurements.

VI. REFERENCES

[1] B.Sc. Graduation Project 1999/2000, VLSI implementation of GPS Digital Correlator. Electronics & Communications Engineering Dept., Ain Shams University.
[2] NAVSTAR GPS. *Public Release Version.* User Equipment, SEPTEMBER 1996.
[3] ICD-GPS-200, Interface Control Document, Rockwell International Corp.
http://navcen.uscg.mil/gps/gpsinfo/gpsdocuments/icd200/icd200c.pdf
[4] KAPLAN, E. (ed.), 1996. Understanding GPS: Principles & Applications. Artech House Publishers, Boston London, 554pp.

[5] SPILKER Jr., J.J. & PARKINSON, B.W. (eds.), 1995. *Global Positioning Systems: Theory & Applications.* American Institute of Aeronautics & Astronautics (AIAA), 1995.
[6]Univ. of Texas. GPS Navigation project:
http://www.utexas.edu/depts/grg/gcraft/notes/gps/gps.html
[7] K. Chadha, "The global positioning system: Challenges in bringing GPS to mainstream consumers," in *Proc. IEEE Int. Solid-State Circuits Conf.*, 1998, pp. 26–28.
[8] M. Simon, J. Omura, R. Scholtz, and B. Levitt, *Spread Spectrum Communications Handbook*. New York, NY: McGraw-Hill, 1994.
[9] R. McDonough and A. Whalen, *Detection of Signals in Noise*. New York, NY: academic, 1995.

VII. ACKNOWLEDGMENTS

We would like to thank Mentor Graphics Egypt, due to their supportive *educational program* to our university. We were able to gain the experience of practicing with the VLSI design tools and this gave us the chance to work in professional circumstances and accelerate our design to a great extent.

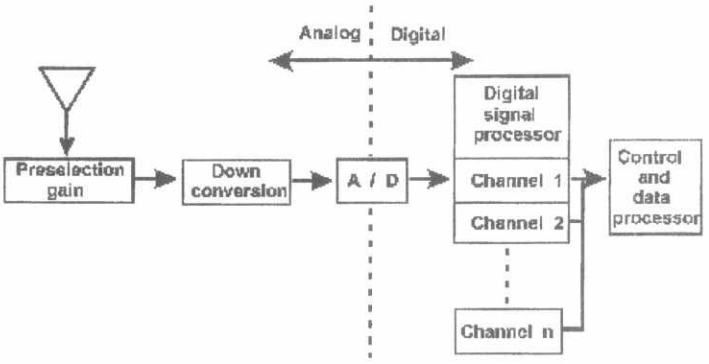

Fig. 1. GPS Digital Receiver System

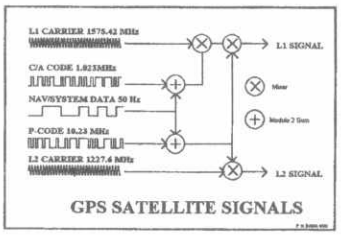

Fig. 2. GPS Signal Components

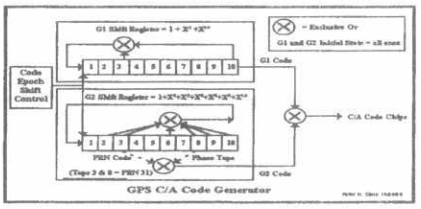

Fig. 3. C/A code generator

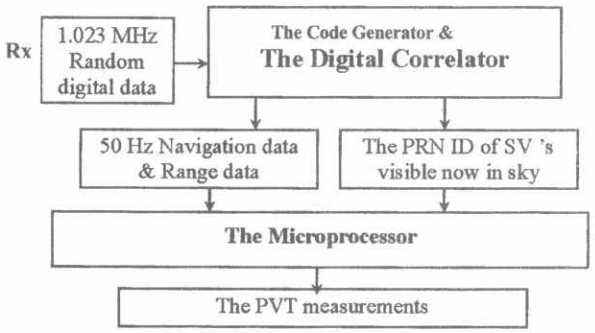

Fig. 4. Digital Correlator's Operation

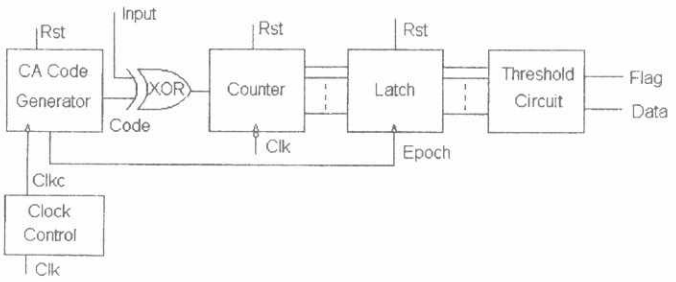

Fig. 5. Serial Correlator block diagram

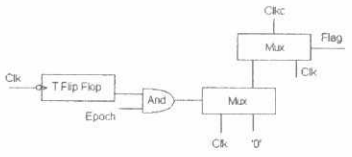

Fig. 6. Clock Control logic

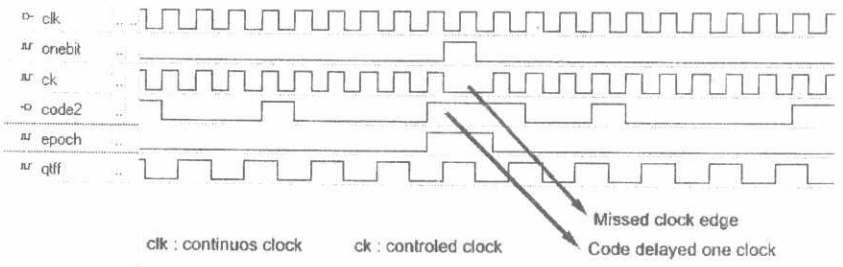

Fig. 7. Simulation results of Serial Correlator

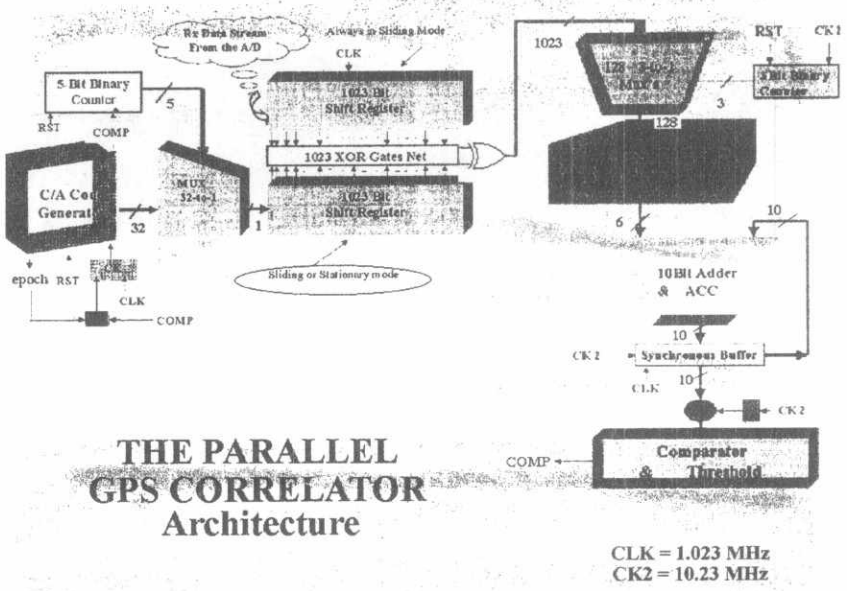

Fig. 8. Parallel Correlator block diagram

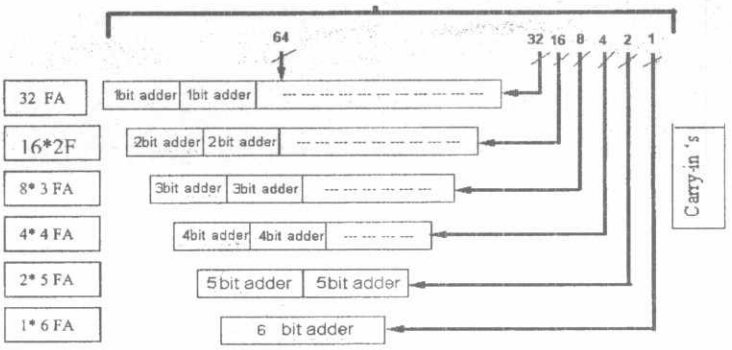

Fig.9. Multi-level Parallel Adder

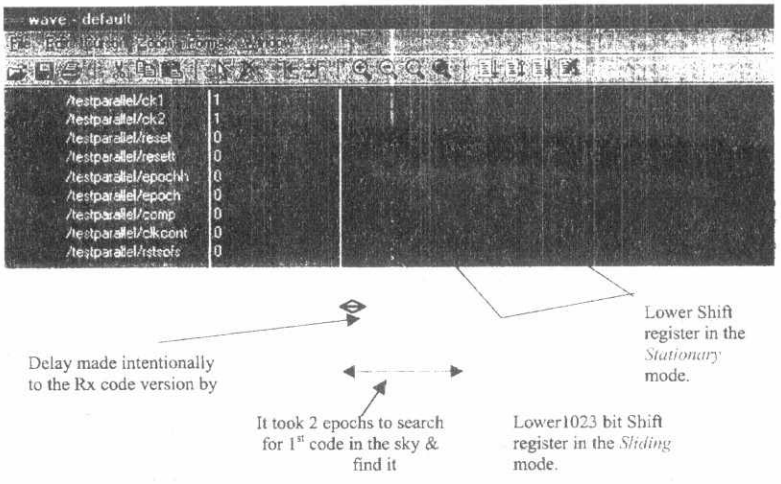

Fig. 10. Simulation results of Parallel Correlator

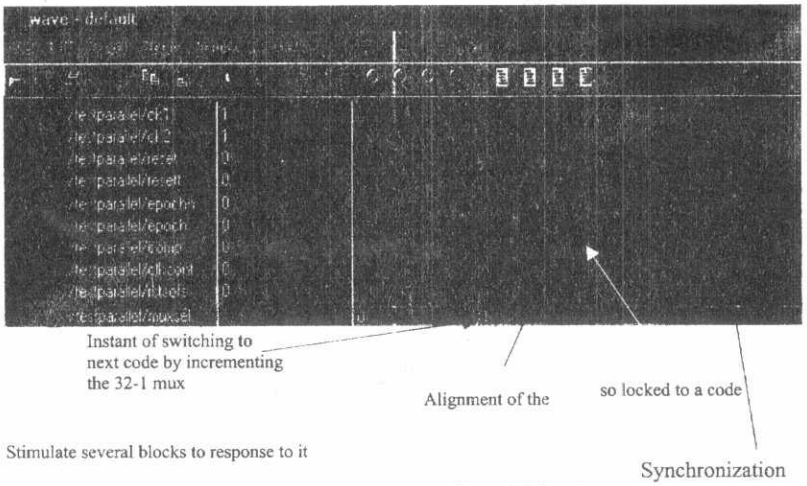

Fig.11. More simulation results of Parallel Correlator

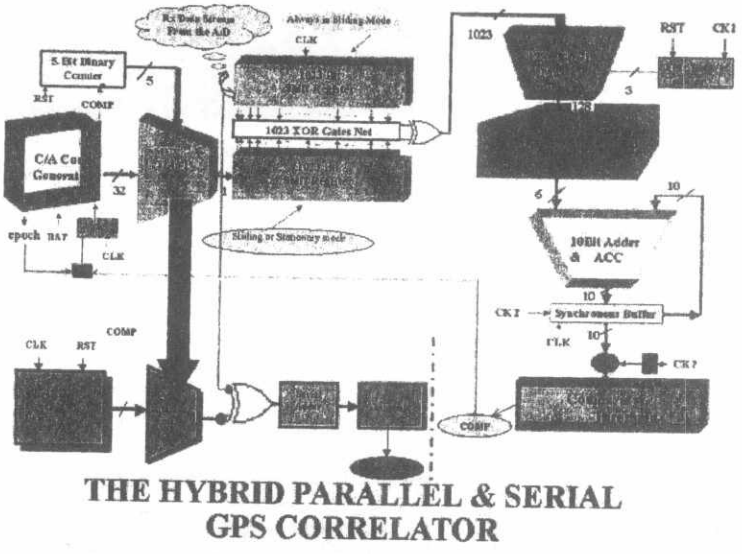

Fig.12. Hybrid Correlator block diagram

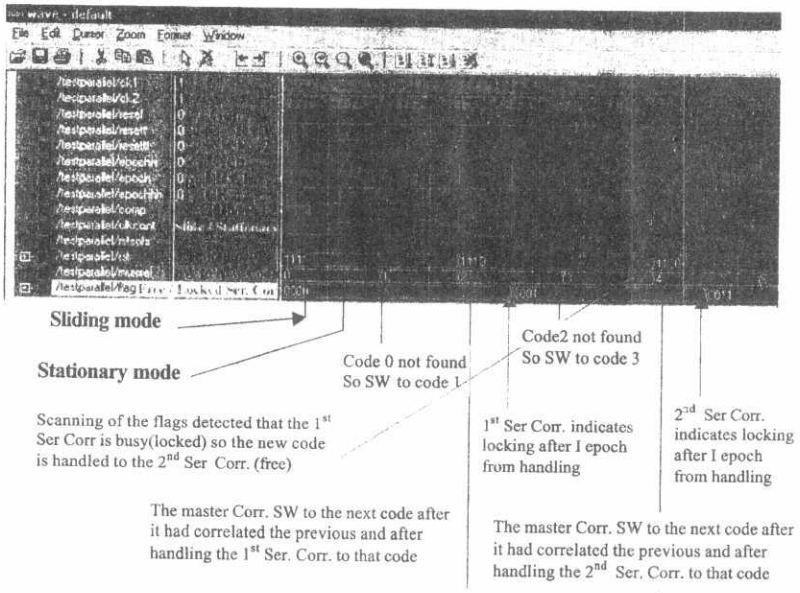

Fig.13. Simulation results of the Hybrid correlator

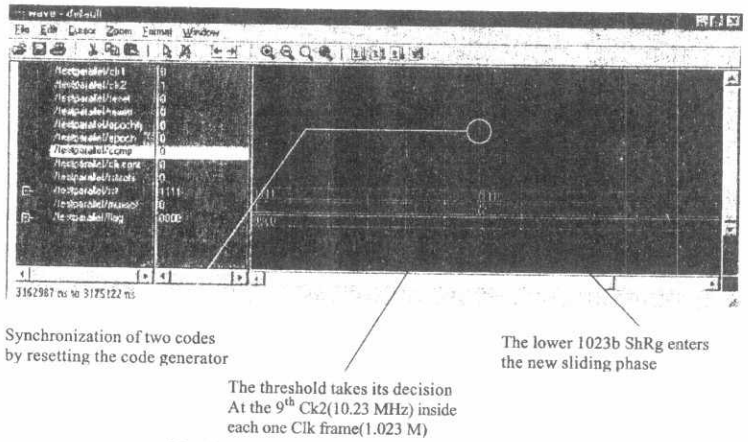

Fig.14. More simulation results of the Hybrid correlator

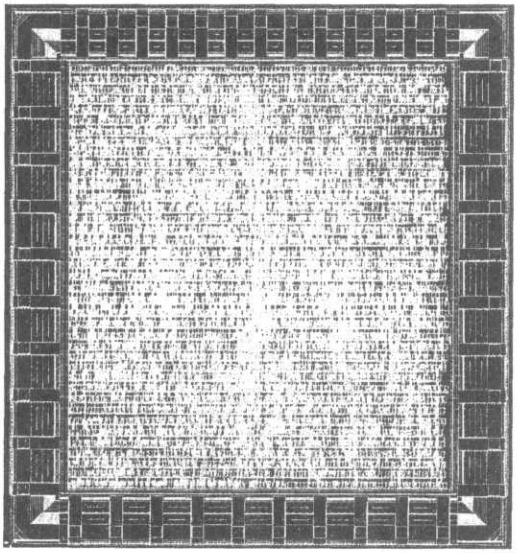

Fig. 15. ASIC Layout of the 4-channel Hybrid correlator using AMS 0.8u technology

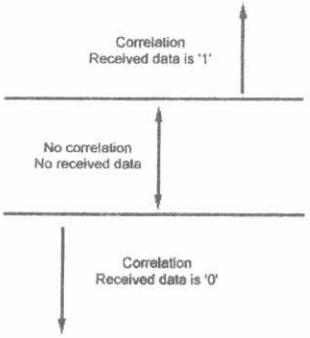

Fig. 16. Threshold Margins of the Correlation