

# A programmable multi-GNSS baseband receiver

Vinh T Tran<sup>1</sup>, Nagaraj C Shivaramaiah<sup>1</sup>, Oliver Diessel<sup>2</sup>, and Andrew G Dempster<sup>1</sup>

School of Electrical Engineering and Telecommunication<sup>1</sup> and School of Computer Science and Engineering<sup>2</sup>

University of New South Wales, Sydney, NSW, 2032, Australia

Email: (vinh.tran, nagaraj, o.diessel, a.dempster)@unsw.edu.au

**Abstract**—This paper assesses the drawbacks in reconfigurability and resource consumption of the conventional baseband signal processing circuitry for modern Global Navigation Satellite Systems (GNSSs) signals. The involvement of new GNSSs leads to the requirement of designing a programmable multi-GNSS baseband receiver that can be reconfigured across GNSS signals. Paralleling baseband circuits is the easiest approach, but resource consumption will be significant. Hardware time multiplexing is more effective technique. It, however, requires an efficient memory hierarchy and an efficient code generator design. These challenges are tackled in the proposed architecture which utilises as 3.2%, 6.6%, 12.5% and 50% resources as the conventional baseband circuitry consisting of 16 GPS L1 C/A channels, 8 BEIDOU B1I channels, 4 GALILEO E1 channels, and 1 GPS L5 channel, respectively. Results are also justified by mathematical analysis.

## I. INTRODUCTION

Beside the Global Positioning System (GPS) and the Russian GLONASS, several new satellite navigation systems are under development: the European GALILEO, Chinese BEIDOU, Japanese QZSS. This creates new challenger in GNSS receiver's design [1], as the receiver should be capable of processing a multitude of signals [2]. This paper analyzes in detail the drawbacks of conventional baseband receiver designed for processing new GNSS signals, especially in resource consumption.

Fig.1 shows a typical GNSS receiver's architecture. Each different frequency band signal is down-converted and sampled by an Analog-to-Digital-Converter (ADC) at the Intermediate Frequency (IF). A standard hardware baseband receiver processes the IF signal and provides accurate estimates of the delay, phase and frequency of the received signal's carrier and spreading code (tracking). It is also often used for the initial coarse estimates of these parameters (acquisition). The processing, usually implemented in software, computes the Position-Velocity-Time (PVT) solution.

The main purpose of acquisition is to acquire visible satellites as quickly as possible. Massively parallel time domain correlators are widely used, however this technique utilises huge resources. The conventional baseband receiver usually operates at the sampling frequency. The FPGA, however, allows digital circuits to run at a much higher speed than the sampling frequency. Considering hardware interleaving technique, this paper proposes a programmable pipeline baseband circuit which not only significantly reduces resource

consumption, but also can be programmed to run as 16 GPS L1 C/A, or 8 BEIDOU B1I, or 4 GALILEO E1, or 1 GPS L5 channels, respectively.

## II. GENERIC CONVENTIONAL BASEBAND ARCHITECTURE

This section describes a generic architecture of a GNSS baseband for signal acquisition and tracking. Its block diagram is illustrated in Fig. 2. The functionality of each block is detailed in [4] and is so not discussed here. The number of  $N$  bit-width of each module (as in Fig. 2) and its effect in the complexity of baseband hardware has been analysed in [3] and is also not further assessed.

Table I shows the center frequency, bandwidth, required IF sampling frequency, primary and secondary code length of new open service signals. The important points to note are: 1) increased signal bandwidths demand higher sampling frequencies, and lead to increased baseband minimum operating frequency; 2) the BOC (Binary Offset Carrier) modulation requires more delayed parallel tap correlator circuits to accurately acquire the signal; 3) use of memory codes demands additional memory to store the spreading codes; 4) use of secondary codes demands a new method to deal with code and data bit transitions.

## III. DRAWBACKS OF CONVENTIONAL ARCHITECTURE

Commercial receivers usually have 12 or more parallel digital baseband receiver channels and most use ASICs design. The disadvantage of ASIC design is specific to one desired GNSS signal and can not be reconfigured as FPGA. Almost of obvious FPGA approaches are first to design a single channel and then multiply it to create a multi-channel receiver (such as [5]). This approach is not ideal as large and expensive FPGAs are required. Considering the relatively low IF sample rates in Table I, interleaving resources in time can be used develop a multi-channel multi-GNSS receiver.

### A. Register-based versus interleaved RAM-based NCO

Numerical Controlled Oscillator (NCO) is an important baseband component. It controls the local replica frequency to align the local generated carrier and spreading code with the input signal. The NCO block diagram is shown in Fig. 3. The output frequency is the input clock frequency divided by the  $f_{\text{control}}$  register's value. Its cycle is completed each time the phase accumulator register overflows. The output frequency resolution is related to the NCO register bit-width. However, generally register based designs are expensive in FPGAs and it is better to efficiently make use of the Xilinx Slices or Altera Logic Elements (noted as FPGA Logic Cells (LCs)). One FPGA LC contains  $\alpha$  m-bit look up tables (LUTs), or  $\beta$   $2^n$ -bit distributed RAMs (Xilinx FPGA) or memory logic

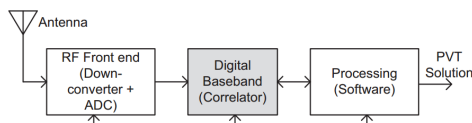


Fig. 1: GNSS receiver block diagram

TABLE I: MODERN GNSS SIGNAL CHARACTERISTICS

Signal name	Carrier frequency (Typical bandwidth) (MHz)	Required IF sampling frequency (MHz)	Modulation type	Primary code length (Memory code ? Y/N)	LFSR length (bits)	Secondary code length (Y/N)	Chipping rate MHz
GPS L1 C/A	1575.42 (2)	4	BPSK	1023 (N)	10	(N)	1.023
GPS L2C	1227.6 (2)	4	BPSK	10230 (N)	27	(N)	1.023
GPS L5	1176.45 (20)	40	BPSK	10230 (N)	13	10 (Y)	10.23
GALILEO E1	1575.42 (4)	8	MBOC	4092 (Y)	N/A	25 (Y)	1.023
GALILEO E5	1191.795 (50)	100	AltBOC	10230 (N)	14	100 (Y)	10.23
BEIDOU B1I	1561.096 (4)	8	QPSK	2046 (N)	11	20 (Y)	2.046

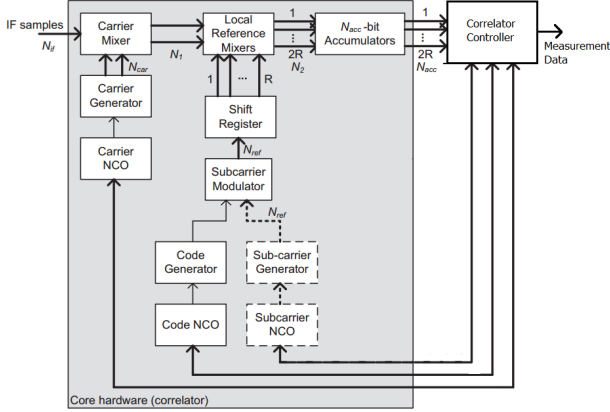


Fig. 2: Generic digital receiver channel block diagram [3]

array blocks (MLAB) (Altera FPGA) allowing  $\beta * 2^n$  bits of data to be stored instead of only  $\lambda$  flip-flops (FF), note that  $\lambda \ll 2^n$  (Fig. 4). An N-bit register-based NCO requires two N-bit register to hold  $f_{\text{control}}$  and the accumulator, an L-bit register to store accumulator latched phase and an N-bit adder; it totally costs  $r = M * (\frac{2N+L}{\lambda} + \frac{N}{\alpha})$  LCs to deploy M NCOs. On the other hand, a combination of a k-bit address counter and a  $2^k \times N$  bit RAM can also be configured as  $2^k$  interleaved N-bit shift registers. The implementation of  $M=2^k$  interleaved N bit RAM-based NCOs, therefore, costs:

$$r = \underbrace{\left\lceil \frac{M}{2^n} \right\rceil * \left( \frac{2N+L}{\beta} \right)}_{\text{RAM-based registers}} + \underbrace{\left\lceil \frac{\log_2(M)}{\lambda} \right\rceil}_{\text{Address counter}} + \underbrace{\frac{N}{\alpha}}_{\text{Adder}} \quad (1)$$

where  $\lceil \cdot \rceil$  is ceiling function. When  $M \geq 2$  this interleaved RAM-based NCO uses much less resources than a register-based NCO.

### B. LFSR code generator versus code memory

As shown in Table I, almost all GNSS spreading codes (except GALILEO E1) can be generated by different length linear feedback shift registers (LFSRs). A multi-GNSS receiver, therefore, needs a combination of various length LFSRs to make up the code generator. It can be multiplexed to generate all of GNSS spreading codes and costs  $r$  LCs:

$$r * \lambda = \underbrace{20 \text{ FFs}}_{\text{GPS L1 C/A}} + \underbrace{22 \text{ FFs}}_{\text{BEIDOU B1}} + \underbrace{26 \text{ FFs}}_{\text{GPS L5}} + \underbrace{28 \text{ FFs}}_{\text{GALILEO E5}} + \underbrace{54 \text{ FFs}}_{\text{GPS L2C}} \quad (2)$$

Besides that, it still requires a 12-bit counter, a 16-1 multiplexer and a 256x16 bit memory to generate 4092-bit GALILEO E1 spreading code (for one desired satellite). FPGA

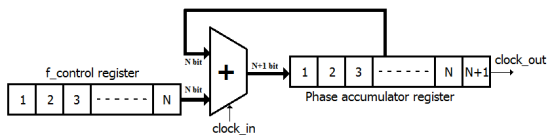


Fig. 3: Generic NCO block diagram

consists of multiple RAM blocks, beside configurable logic blocks. The memory module, thus, can be deployed as a distributed RAM or RAM blocks. As a result, the total resource consumption is:

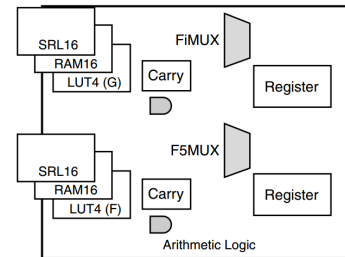
$$r = \left\{ \begin{array}{l} \underbrace{\frac{150}{\lambda}}_{\text{LFSRs}} + \underbrace{\frac{12}{\lambda}}_{\text{12 bit counter}} + \underbrace{\left\lceil \frac{16}{m} \right\rceil}_{\text{16-1 multiplexer}} + \underbrace{\left\lceil \frac{256}{2^n} \right\rceil * \frac{16}{\beta}}_{\text{256x16 bit distributed RAM}} \quad (\text{LCs}) \\ \underbrace{\frac{150}{\lambda}}_{\text{LFSRs}} \text{ LCs} + \underbrace{\frac{12}{\lambda}}_{\text{12 bit counter}} \text{ LCs} + \underbrace{\left\lceil \frac{16}{m} \right\rceil}_{\text{16-1 multiplexer}} \text{ LCs} + \underbrace{1}_{\text{256x16 bit}} \text{ Block RAM} \end{array} \right. \quad (3)$$

This is only the resource required for one multi-GNSS code generator. If the baseband receiver has M channels, the resource consumption is at least M times of the indispensable resource. Instead of LFSR, the code generator can be designed by the RAM-based prototype, which includes a counter and a memory block whose size should be enough to store the longest spreading code (10230 bits). A 14-bit counter and a 1024x16 bit distributed/block RAM is the solution. The block RAM prototype is preferable. However, the design is still wasteful of resources, especially when implemented the short spreading codes. Take GPS L1 C/A, BEIDOU B1I, or GALILEO E1 as an example, only 1/16, 1/8, or 1/4 memory resources are used, respectively.

The spreading code chipping rate is usually slow ( $\leq 10\text{MHz}$ ), so that, hardware time interleaving can be used to reduce utilized resources. The M channel multiplexed k-bit counter ( $k=\log_2(\text{code\_length})$ ) can be mapped on only  $r$  LCs, and spreading codes are stored in  $\left\lceil \frac{M * \text{code\_length}}{1024 * 16} \right\rceil$  1024x16 bit RAM blocks, with r is:

$$r = \underbrace{\left\lceil \frac{M}{2^n} \right\rceil * \frac{k}{\beta}}_{\text{Counter RAM}} + \underbrace{\left\lceil \frac{M}{2^n} \right\rceil * \frac{k}{\beta}}_{\text{Counter Max RAM}} + \underbrace{\frac{k}{m}}_{\text{k-bit Comparator}} + \underbrace{\frac{k}{\alpha}}_{\text{k-bit Adder}} \quad (4)$$

As a result, 16 GPS L1 C/A, or 8 BEIDOU B1I, or 4 GALILEO E1, or 1 GPS L2/L5 or 1 GALILEO E5 code generators can synthesise on the same resource. Taking Xilinx Spartan-3 FPGA as an analysed platform, the combination of the time interleaved technique and code memory reduce resource consumption to maximum M-times (M short spreading


 Fig. 4: Xilinx SLICEM block diagram ( $\alpha=\beta=\lambda=2, m=n=4$ )

code channels) and minimum 58% LC consumption (M long spreading code channels).

### C. Active parallel correlator versus passive matched filter correlator

Parallel code search can be either Active Parallel Correlator (APC) or Passive Matched Filter (PMF) type. The conventional baseband receiver is developed based on APC as in Section II. The advantage of the APC is the baseband receiver can run at the IF sampling frequency. Every received signal is correlated with multiple locally generated code delay taps. It therefore achieves the maximum signal to noise ratio (SNR) and is easily implemented. However the scalar parallel architecture consumes significant resources. An R-tap APC as in Fig. 2 consumes  $r$  LCs:

$$r = \underbrace{\left\lceil \frac{R}{\lambda} \right\rceil}_{\text{R-tap Shift Register}} + \underbrace{\frac{2R}{\alpha} * \left\lceil \frac{N_1 + N_{ref}}{m} \right\rceil}_{\text{2R (N}_1+N_{ref}\text{)-bit LUTs Local Reference Mixer}} + \underbrace{2R * \left( \left\lceil \frac{N_{acc}}{\lambda} \right\rceil + \left\lceil \frac{N_{acc}}{\alpha} \right\rceil \right)}_{\text{2R } N_{acc}\text{-bit Accumulators}} \quad (5)$$

Moreover, the APC takes a long time to acquire satellites, if incoming data includes a bit transition caused by a navigation data or secondary code [6]. The APC has to search for the start bit of the primary code and so on the multiple code delayed tap technique does not reduce acquisition time.

The PMF is introduced in [7] and a modified version, that is more efficient digital logic design, called the Passive Parallel Correlator (PPC) [8]. In the PPC, the baseband sampled signals are passed through a tapped delay-register. Each tap is then multiplied by a stationary local reference code and summed using a binary tree network. The bit-width of the tapped delay-register is the number of parallel executed correlators. The disadvantage of the PPC is it has to process at an integer-time of the chipping rate to have the same delayed correlators as APC. The SNR also decreases as a result. The receiver performance, however, does not degrade after appropriate filtering and resampling the baseband signal to exactly a number of samples per code chip [9]. The main advantage of PPC is that all R-tap correlators correlate from the primary code's first bit so that the problem of bit transition is avoided. The PPC also allows sharing RAM blocks to store the accumulators' value. The resource consumption of a R-tap PPC that consists of  $N_1$ -bit Carrier Mixer,  $N_{ref}$ -bit Local Reference signal,  $N_2$ -bit Local Reference Mixer and  $N_{acc}$ -bit Accumulators, however, is still high as:

$$r = \underbrace{(2 * N_1 + N_{ref}) \left\lceil \frac{R}{\lambda} \right\rceil}_{\text{(2*N}_1+N_{ref}\text{) R-bit registers}} + \underbrace{\frac{2R}{\alpha} * \left\lceil \frac{N_1 + N_{ref}}{m} \right\rceil}_{\text{2R (N}_1+N_{ref}\text{)-bit LUT Mixer}} + \underbrace{\left\lceil \frac{N_{acc}}{2} \right\rceil}_{\text{N}_{acc}\text{-bit Adder}} + \underbrace{2 * \sum_{i=1}^{\log_2 R} \left\lceil \frac{R}{2^i} \right\rceil \left\lceil \frac{i + N_2 - 1}{2} \right\rceil}_{\text{2 Tree Network Adder}} + \underbrace{2 * \left\lceil \frac{R}{K} \right\rceil}_{\text{KxN}_{acc}\text{ bit RAM block}} \text{ blocks} \quad (6)$$

Consequently, if the receiver has M channels, the resource consumption of APC and PPC is at least M-times of  $r$ . Registers and parallel accumulators are the most consumed resources. The PPC architecture logically allows LUT mixers, tree network adders and accumulators to be multiplexed among channels. This paper, therefore, proposes the interleaved PPC, where all arrays of registers are replaced by distributed RAM.

Combining with hardware time interleaving and efficient memory hierarchy, resource utilization can reduce significantly. The M channel interleaved PPC now costs only  $r^*$  resource.

$$r^* = \underbrace{\left\lceil \frac{M * (2 * N_1 + N_{ref})}{2^n} \right\rceil \left\lceil \frac{R}{\beta} \right\rceil}_{\text{M*(2*N}_1+N_{ref}\text{) R-bit RAM-based registers}} + \underbrace{\left\lceil \frac{2R}{\alpha} \right\rceil * \left\lceil \frac{N_1 + N_{ref}}{m} \right\rceil}_{\text{2R (N}_1+N_{ref}\text{)-bit LUT Mixer}} + \underbrace{\left\lceil \frac{N_{acc}}{2} \right\rceil}_{\text{N}_{acc}\text{-bit Adder}} + \underbrace{2 * \sum_{i=1}^{\log_2 R} \left\lceil \frac{R}{2^i} \right\rceil * \left\lceil \frac{i + N_2 - 1}{2} \right\rceil}_{\text{2 Tree Network Adder}} + \underbrace{2 * \left\lceil \frac{M * R}{K} \right\rceil}_{\text{KxN}_{acc}\text{ bit RAM block}} \text{ blocks} \quad (7)$$

Table II shows Xilinx Spartan-3 resource consumption of M=16 channels of the three architectures APC, PPC, and interleaved PPC that can correlate all GNSS signal types with R=32 taps,  $N_1=4$  bits,  $N_{ref}=1$  bit,  $N_2=5$  bits,  $N_{acc}=32$  bits and K=512. The proposed interleaved PPC has the lowest resource consumption.

## IV. REPROGRAMMABLE PIPELINE BASEBAND RECEIVER

### A. Proposed baseband receiver architecture

We now propose a pipeline interleaved baseband receiver as shown in Fig. 5. The receiver consists of 7 pipeline stages and a 32-tap interleaved PPC. Every module separated into pipeline stages satisfies that it is no more than one memory access block in each stage. The delta delay time is also considered. The most time delayed component is the tree adder. It contains 5 combination logic layers, and takes more than 13 ns to process (estimated by Xilinx ISE 13.3 on Spartan 3 FPGA). It is, therefore, separated into 2 stages (5<sup>th</sup>, 6<sup>th</sup> stage) to fulfil the timing constraints. The baseband operating frequency is configured as M times the IF sampling frequency, where M is the number of interleaved channels. The circuit is implemented on Xilinx Spartan FPGA and can run at high speed clock: 160 MHz on Spartan-3 and 200 MHz on Spartan-6. It is enough to process multiple GNSS baseband channels which usually operate at the IF sampling frequency. According to Table I, the baseband circuit can be programmed to run as 16 GPS L1 C/A, 8 BEIDOU B1I, 4 GALILEO E1, or 1 GPS L5/ GALILEO E5 channels, respectively. The channels are selected through the Channel Address bus controlled by an external 4-bit counter. The counter maximum value is also programmed, thus the baseband receiver can configure as a variety of channels. Besides that, each individual channel can be accessed and controlled by the correlator controller via its dedicated address. It, therefore, allows the interleaved channels can similarly operate as the corresponded parallel channels.

1) **CARRIER NCO and CODE NCO:** Both Carrier NCO and Code NCO are designed as a 32-bit RAM-based NCO, with phase accumulator and phase dump are single port RAMs,  $f_{control}$  is a dual port RAM that allows the correlator controller to change individual channel's  $f_{control}$  while the NCO is running.

2) **CODE COUNTER:** is a 14-bit interleaved RAM-based counter, it includes a counter value RAM, a counter maximum RAM and an 14-bit adder. The counter maximum value RAM

TABLE II: Utilized resource of three correlator architectures

Architecture	4 input LUTs	Flip-flops	Occupied slices	18Kbit RAM blocks
APC	34816	33280	34048	0
PPC	7680	4608	6144	32
Interleaved PPC	768	0	384	2

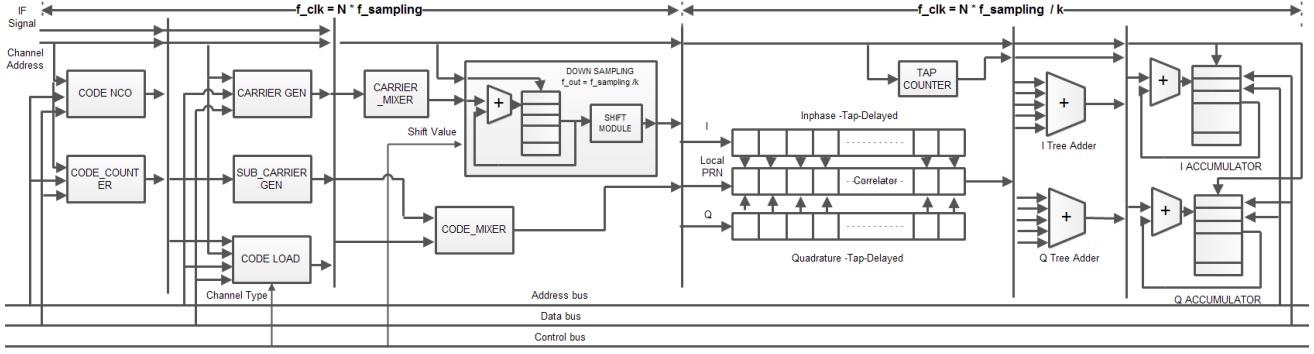


Fig. 5: Pipeline interleaved baseband hardware receiver

can be programmable by the correlator controller, and the counter value is reset to zero when it exceeds the counter maximum value. The counter, therefore, can be configurable to count every value smaller than  $2^{14} - 1$ , it is enough for all of GNSS primary code's length.

3) **CODE LOAD**: contains a 512x32 bit code memory block (one 18 Kbit block RAM) and a 32-1 multiplexer. The correlator controller controls the Channel Type signal to multiplex the allocation of the Channel Address bus and the Code Counter value to make up the address of the code memory ( $n$  bits of the Channel Address is  $n$  MSBs and  $m$  bits of the Code Counter value is  $m$  LSBs ( $n + m = 9$ )). One FPGA block RAM, as a result, can be separated as 16 GPS L1 C/A 1023 bit code memory blocks ( $n=4, m=5$ ), or 8 BEIDOU B1I 2046-bit code memory blocks ( $n=3, m=6$ ), or 4 GALILEO E1 4092-bit code memory blocks ( $n=2, m=7$ ).

4) **DOWN SAMPLING**: downsamples the baseband signal from the IF sampling frequencies an integer time for spreading code chip rate. The PPC resampling filter [9] is complicated and increases the bit-width of the output data, it leads to higher complexity of the correlator. Therefore we propose a new resampling module that downsamples the baseband signal by to  $2^i$ -division of the sampling frequency. The output frequency is the nearest frequency higher than a desired multiple of chipping rate (2 for BPSK signal and 4 for BOC signal). The output signal is the modulo  $k = 2^i$  of  $k$  consecutive baseband signals, thus the data bit-width remains. The  $i$  value can be configured via the Shift Value signal by Correlator Controller. It also, as a result, allows the correlator, tree adder, and accumulator stage to run at a  $k$  times slower clock speed (Fig.5).

5) **CORRELATOR**: is a 32-tap interleaved PPC analysed in Section III-C.

6) **ACCUMULATOR**: contains 32-bit adder and a 18 Kbit RAM block. The interleaved technique and the combination of the Channel Address and the Tap Counter value are used to allocate the RAM block's address. It allows one 18 Kbit RAM block to be shared to store 32 correlator values of  $M$  channels ( $M=1,4,8,16$ ).

### B. Resource consumption

Table III shows the Xilinx Spartan 3 FPGA resource consumption of the proposed baseband circuit in comparison with a conventional baseband circuit that contains 16 GPS L1 C/A, or 8 BEIDOU B1I, or 4 GALILEO E1, or 1 GPS L5 channels, respectively. These conventional channels are implemented with an  $R=32$ -tap APC and  $N_1=4$  bits,  $N_{ref}=1$  bit,  $N_2=5$

TABLE III: SPARTAN 3 RESOURCE UTILIZATION

Baseband circuit (n channels)	4-input LUTs	Flip-flops	Occupied slices	18 Kbit RAM blocks	Operating frequency
16 GPS L1 C/A	36160	36672	36416	0	4 MHz
8 BEIDOU B1I	18088	18352	18220	0	8 MHz
4 GALILEO E1	9848	9136	9492	0	8 MHz
4 GALILEO E1	9128	9136	9132	4	8 MHz
1 GPS L5	2264	2300	2282	0	40 MHz
Proposed receiver	2227	790	1170	3	128 MHz

bits,  $N_{acc}=32$  bits. The result combining with Eq.1, Eq.3-7 reveal that the proposed baseband receiver can reduce resource utilization dramatically while meeting the timing requirements of each receiver.

### V. CONCLUSION

The drawbacks in configurability and resource consumption of the conventional baseband circuits were assessed for multiple modern GNSS signals. APC, PPC and the proposed interleaved PPC were compared. Mathematical analysis affirms that the proposed PPC has the lowest resource utilization and is capable of processing new modern GNSS signals. A programmable pipeline baseband circuit is proposed. Relying on hardware time interleaving and the efficient memory hierarchy, the proposed baseband circuit can be configured to run as 16 GPS L1 C/A, or 8 BEIDOU B1I, or 4 GALILEO E1, or 1 GPS L5 receiver channels with 3.2%, 6.6%, 12.5%, and 50% resource consumption of the corresponding conventional baseband circuits.

### REFERENCES

- [1] A. Dempster, "Satellite navigation: New signals, new challenges," in *ISCAS. Circuits and Systems IEEE International Symposium on*, May 2007, pp. 1725–1728.
- [2] A. G. Dempster and S. Hewitson, "The system of systems receiver: An Australian opportunity," in *IGNSS Symposium on GPS/GNSS*, 2007.
- [3] N. C. Shivaramaiah and A. G. Dempster, "On the baseband hardware complexity of modernized gnss receivers," in *Circuits and Systems, Proceedings of IEEE International Symposium on (ISCAS)*. IEEE, 2010, pp. 3565–3568.
- [4] E. D. Kaplan and C. J. Hegarty, *Understanding GPS: principles and applications*. Artech house, 2005.
- [5] P. J. Mumford, K. Parkinson, and A. G. Dempster, "The namuru open gnss research receiver," in *Proceedings of the ION ITM*, 2006, pp. 2847–2855.
- [6] J. B.-Y. Tsui, *Fundamentals of Global Positioning System Receivers-A Software Approach*. A John Wiley & Sons Inc Publication, 2000, vol. 2.
- [7] A. Polydoros and C. Weber, "A unified approach to serial search spread-spectrum code acquisition—part ii: A matched-filter receiver," *Communications, IEEE Transactions on*, vol. 32, no. 5, pp. 550–560, May 1984.
- [8] W. Namgoong and T. Meng, "Power consumption of parallel spread spectrum correlator architectures," in *Low Power Electronics and Design. Proceedings. International Symposium on*, Aug 1998, pp. 133–135.
- [9] W. Namgoong, S. Reader, and T. Meng, "An all-digital low-power if gps synchronizer," *Solid-State Circuits, IEEE Journal of*, vol. 35, no. 6, pp. 856–864, June 2000.