3GPP Searcher v1.0

Introduction

The 3GPP Searcher core is a highly integrated solution for identifying the multiple transmission paths of users in a 3GPP uplink. The core includes all the logic required for scramble code generation, correlation, accumulation and filtering in a single co-processor, easily integrated with a DSP or microprocessor.

Features

- Device families supported: Virtex™-4, Virtex-5, Spartan™-3A DSP
- Scalable solution for femto-cells to macro-cells
- Algorithm features:
  - Correlation against pilot bit and data bits in DPCCH channel
  - Multiple search correlations in parallel
  - Filtered and unfiltered PDP generation
- High level of integration, encapsulating all circuitry required to generate and maintain PDPs for each search:
  - Automatic scramble code advance
  - Scramble code and pilot generation
  - Search correlation, coherent and non-coherent accumulation
  - PDP filtering
- Design scales with following parameters to minimize resource utilization, based on:
  - Number of searches, results, and antennas
  - Oversample and clock rates
  - Window size
  - Quantization
  - Scheduling period
- Designed for efficient scheduling of searches:
  - Fast changing channels can be scheduled more frequently than slower changing channels, optimizing hardware resources
  - Pipelined firmware operation — new search configurations written concurrently while performing current searches and reading results of previous searches.
  - Design automatically advances scramble code for each search
- Easy integration to microprocessor/DSP via OCP-compatible interfaces
  - Pipelined read of search results for speed
  - DMA request interrupt on search configuration interface enables DMA transfers of search configurations
- For use with Xilinx CORE Generator™ v9.2i or later

System Overview

Figure 1 shows a typical use of 3GPP Searcher core. The core is designed to act as a co-processor attached to a microprocessor or DSP across a system bus. The open core protocol (OCP) compatible interfaces allow easy adaptation to other bus protocols.

During operation, the processor writes a block of search configurations to the 3GPP Searcher core. This configuration details scrambling code, delay, slot format and associated data. At the start of a search period, the Searcher takes a block of search configurations, and for each search, starts correlating against the incoming antenna data stream. The antenna data stream can come directly from a radio interface, but could also be streamed by DMA across the system bus.

At the end of the search, the resultant power delay profiles (PDPs) for each search can be read across the search results interface on to the system bus.

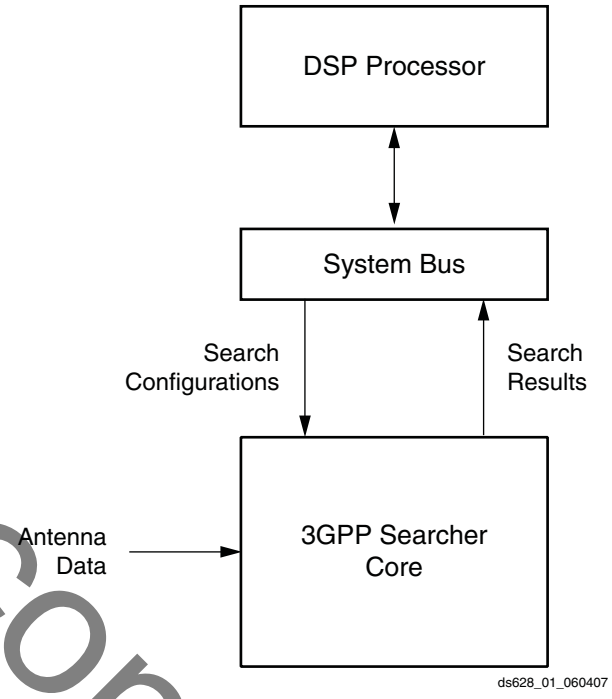


Figure 1: **Typical Application**

Background

Searching for Multiple Paths

The 3GPP Searcher core is used to search for and track the multiple transmission paths from various user equipment (UEs) to the basestation (BS) in a 3GPP uplink. As an example, a simple radio channel environment with multiple paths is shown in Figure 2. Correlating the received signal at the BS against that transmitted by the UE gives a power delay profile for the UE. This profile changes as the UE moves through the environment.

The Searcher core is given an initial estimate of the channel delays from the 3GPP random access channel (RACH) detection block which detects the initial presence of a UE. This estimate is used to set the offset (window delay) of the search window that the Searcher core uses to find a particular UE (Figure 2).

The Searcher core operates by correlating against the uplink dedicated physical control channel (DPCCH) sent by each UE. The result of the correlation represents a PDP for each UE. The PDP has a number of peaks corresponding to the different transmission paths in the radio channel.

The example PDP shown in Figure 2 is simplified to show just the peaks for each path. In contrast, an actual PDP generated by the Searcher core when compared with a reference model is shown in Figure 3.

Figure 3 shows four PDPs generated when searching for four different UEs. The output PDP from the core is shown in red. For comparison, results from a reference model are shown in blue for a complete range of delays. In a real channel, paths undergo fading as a UE moves through the environment. To combat this effect, a time-weighted filtering is typically applied to get a longer term average for a UE. This average can be used to identify the main transmission paths, and configure the RAKE receiver in a 3GPP WDCMA basestation.

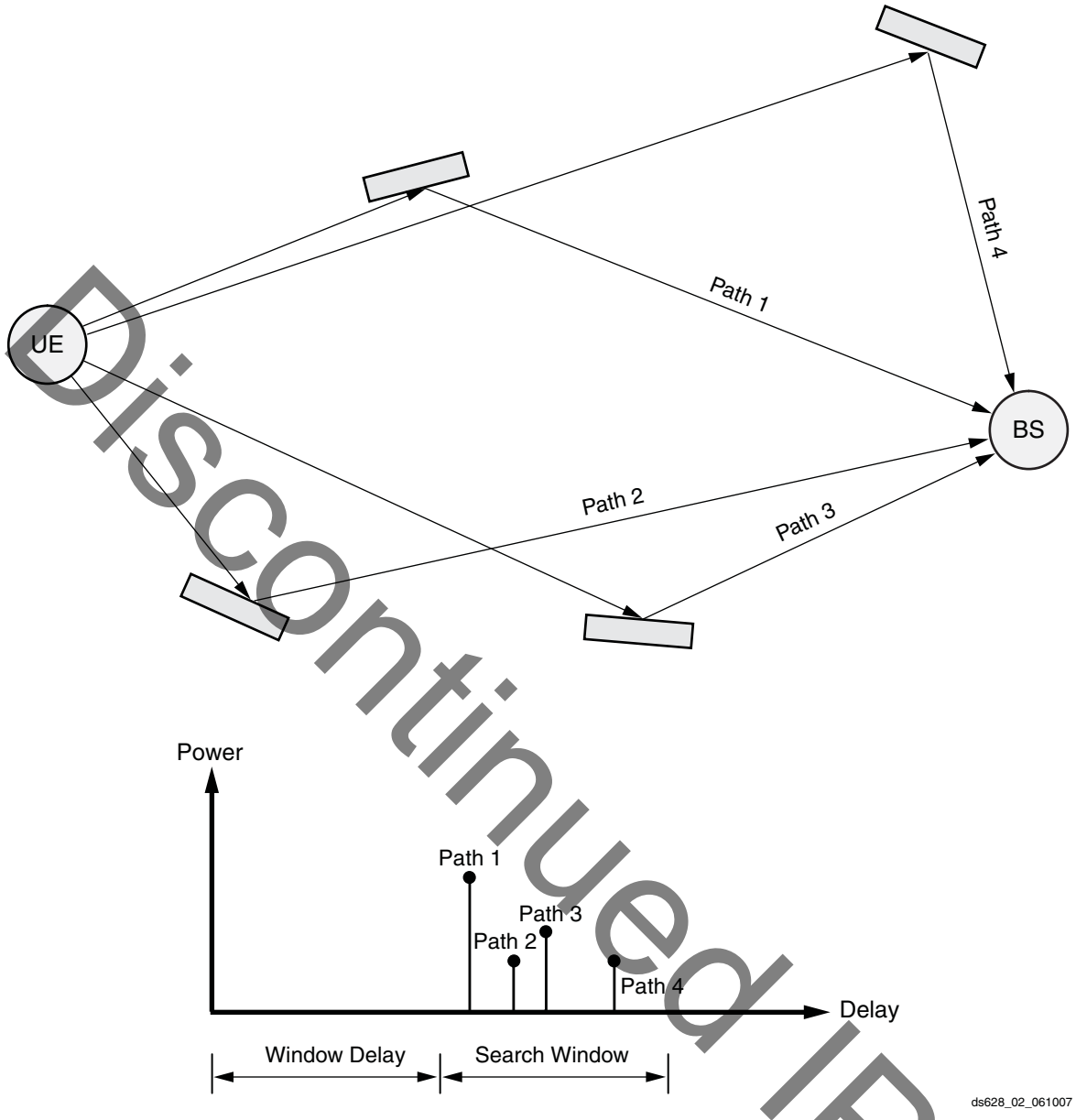


Figure 2: **Simple Radio Channel Environment and Power Delay Profile**

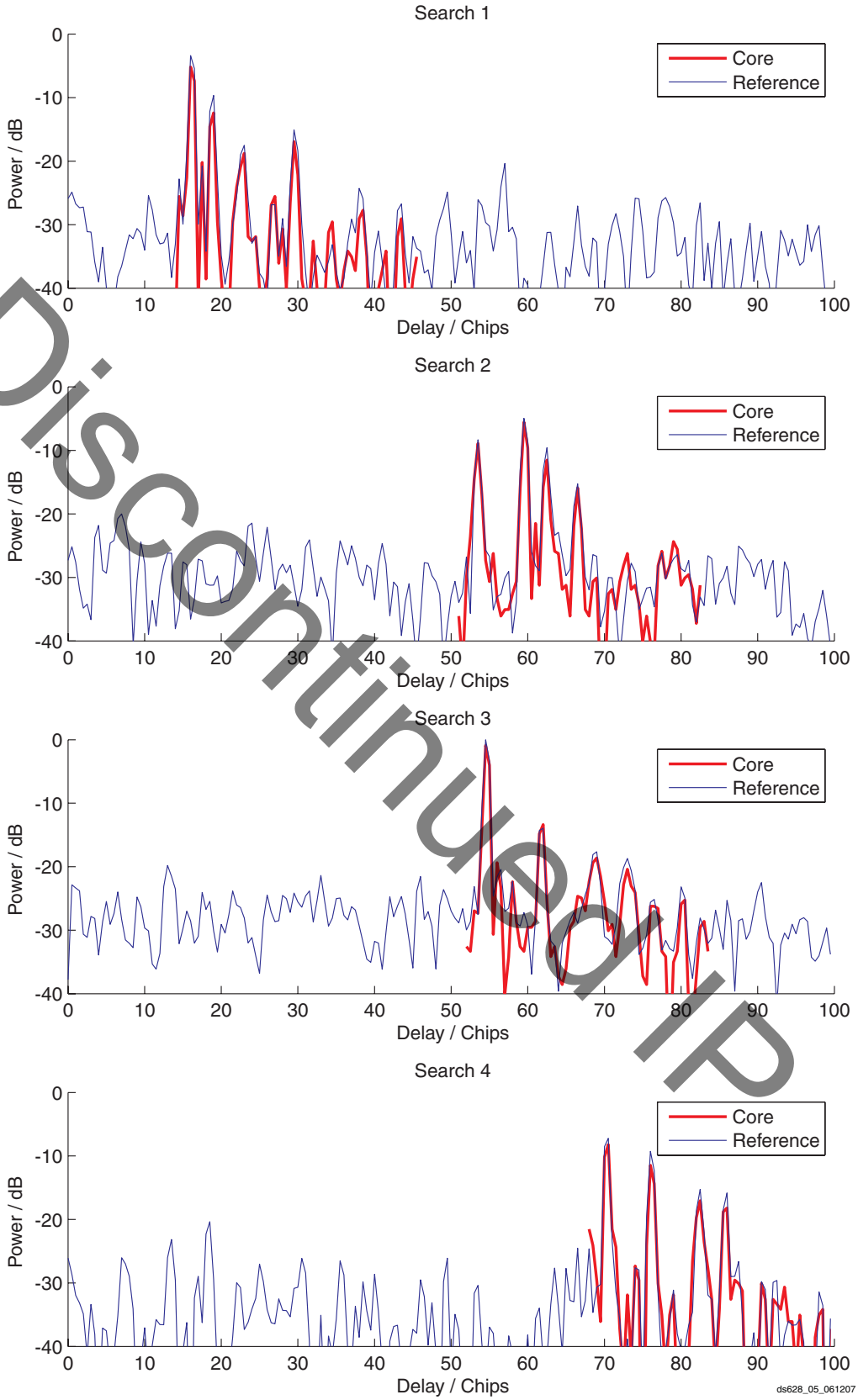


Figure 3: **PDPs Generated by Searcher Core**

DPCCH Format

Figure 4 shows the slot format of the DPCCH channel on the uplink. Each radio frame is composed of 15 slots, with each slot on the DPCCH consists of 10 bits. The DPCCH has a spreading factor of 256, so each bit is composed of 256 chips.

Each bit in Figure 4 is shown broken into a number of windows. These windows are the basic unit of operation of the search, and determine the length of the PDP generated by the Searcher core. For each window, the correlator block in the Searcher core generates a correlation result. The results for a window are then accumulated coherently, or non-coherently, depending on the pattern of bits in the DPCCH slot (see section 5.2 of 3GPP TS 25.211 V6.7.0, *Physical channels and mapping of transport channels onto physical channels (FDD) (Release 6)*, for further details).

As shown in Figure 4, each DPCCH slot contains 3 to 8 pilot bits, dedicated to providing a known sequence for the Searcher core to correlate on. As well as these pilot bits, the DPCCH contains other variable-length control fields: transport format combinator indicator (TFCI), feedback indicator (FBI), and transport power control (TPC).

The 3GPP Searcher core block correlates on both pilot and data bits in the DPCCH. This correlation is performed coherently across known bit sequences in the DPCCH. Coherent correlation involves accumulating results separately for the I and Q components of the antenna data stream and has the benefit of reducing noise on the I and Q accumulated results. Coherent results for each known sequence of data are then accumulated non-coherently by accumulating the correlated power, weighted by the length of the coherent correlation.

Sections of coherent correlation are marked with shaded sections in Figure 4. The pilot bits follow a known sequence according to the slot format and number, forming one block for coherent correlation. The TFCI and FBI bits do not contain a known data sequence, so can only be correlated against across single bits. Finally the TPC field can only take the values `00` and `11`, allowing coherent correlation for 2 bits.

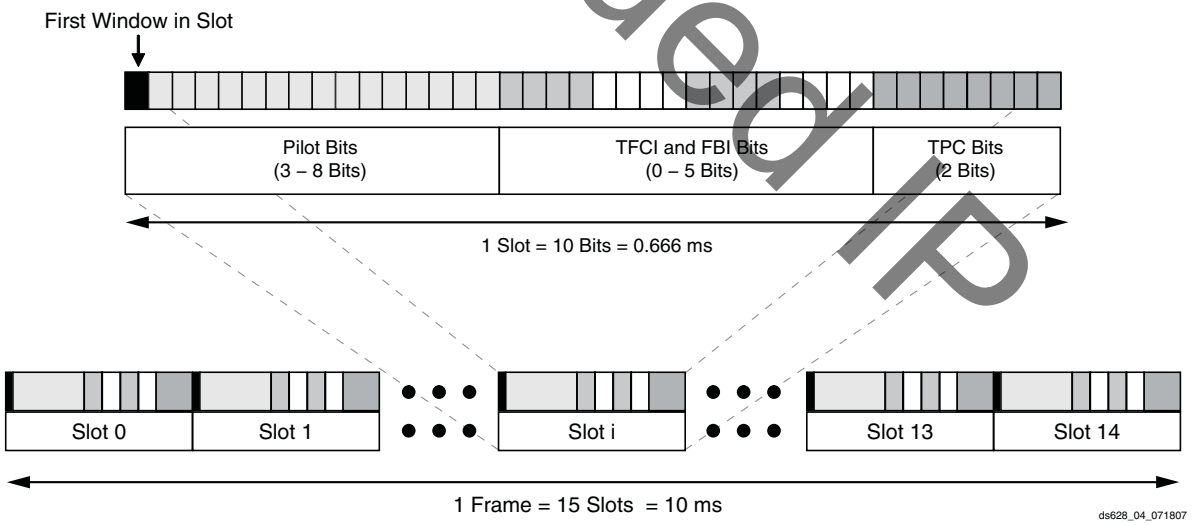


Figure 4: **DPCCH Slot and Frame Format**

System Operation

System Overview

Figure 5 shows a block diagram of the 3GPP Searcher core, consisting of a five-stage processing pipeline, and a control count block. This block controls the data flow through the pipeline and synchronizes the core to the incoming antenna stream.

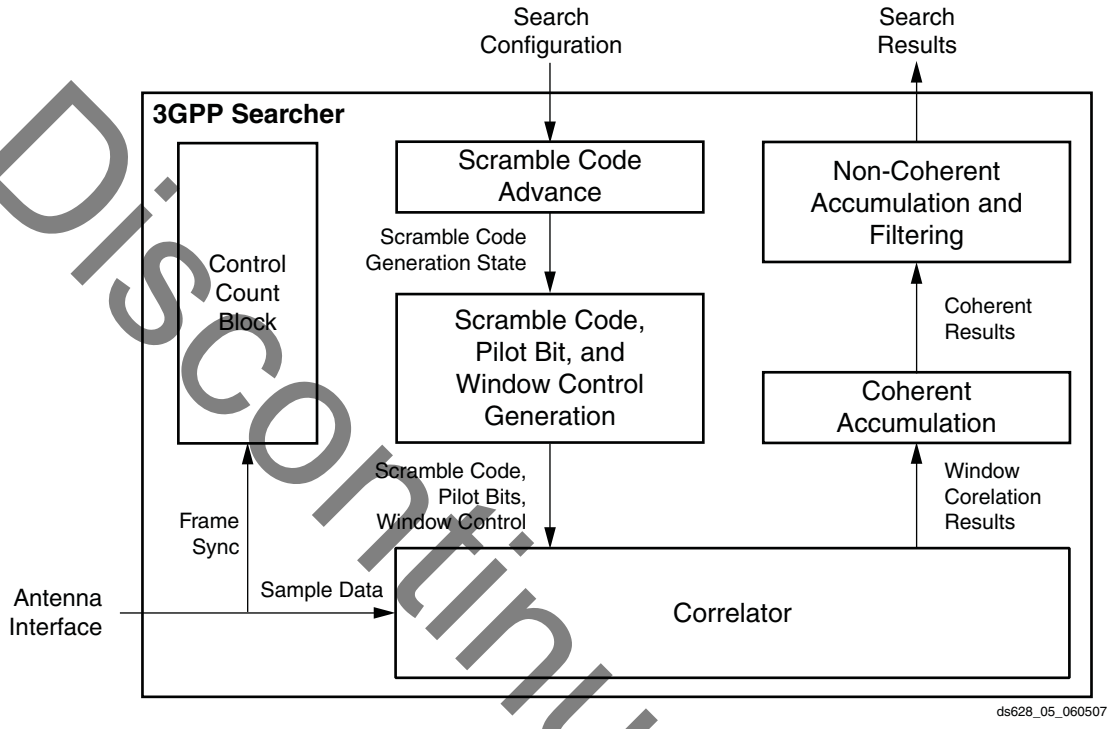


Figure 5: **3GPP Searcher Core Block Diagram**

Before data processing begins, a block of search configurations is written into the scramble code advance unit at the start of the pipeline. Search configurations consist of a number of fields including the scrambling code, antenna, offset and number of pilot bits used for each search.

The first stage of data processing pipeline is the scramble code advance block. This block advances the scrambling code for each search, adjusting for the search window offset and the start time of the next search period. The output of this block is a the scramble code generation state for each search. This output is loaded into the next stage at the start of a search period.

The second stage in the pipeline is the scramble, pilot and control generation block. This block generates the scrambling code and pilot sequence for each search, together with control signals for downstream blocks. These signals control when results are passed from the coherent to the non-coherent accumulation stages.

The third stage of the pipeline is the correlator. This stage correlates the scrambling code sequence for each search generated by the previous block and correlates against sample data from the appropriate antenna data stream. The correlator block generates one result for each delay in the search window, feeding these results in to next block.

The fourth pipeline stage accumulates the correlation results coherently for I and Q results, passing these results on the final block in the pipeline.

The final pipeline stage is non-coherent accumulation and filtering. This stage converts I and Q coherent correlation results to power and scales the results to account for the different lengths of the coherent accumulation. The results can either be accumulated for an instantaneous PDP or passed through a time-weighted filtering stage. When the block of searches is complete, the results can be read by the processor across the search results interface.

The subsequent sections describe the operation of the 3GPP Searcher core in more detail.

Detailed Operation

Control Count Block

The control count block synchronizes the data processing pipeline operations to the incoming antenna data streams. Figure 6 shows how each processing clock cycle in the system is identified by a hierarchy of counts. Shading indicates how these counts map on to the fields within the DPCCH described in "DPCCH Format" on page 5.

The basic unit of transmission in 3GPP W-CDMA is the chip. The 3GPP Searcher core clock operates at a multiple of the chip rate as shown in the top set of counts in Figure 6. Each chip consists of *O* samples, with *L* processing cycles available per sample. Thus, the Searcher core operates at *OL* times the chip rate.

Both *O* and *L* are related to the core parameters: *O* is the oversample rate parameter, and *OL* is the clock rate parameter.

Within a chip, the cycle count identifies samples from different antennae. Data for antenna 0 is expected on cycle count 0, antenna 1 on cycle count 1, etc. If data is unavailable, then the core stalls and does not process data on that cycle. See "Timing Diagrams" on page 22 on interface timing for more details.

The next level in the count hierarchy is involved in counting chips within a 3GPP slot. A slot is composed of 2560 chips which are identified by the chip count. The chip count can be broken into a number of sub-counts. The bit count identifies each bit within the DPCCH slot. Within each bit, the window count identifies each window of data processed by the correlator block The window size is variable and can be set by the user via the PDP_Length core parameter. In the example shown in Figure 6, the window size is 64 chips, resulting in 4 windows per bit.

The final level of count hierarchy involves a 3GPP frame. Each frame is composed of 15 slots, with each frame identified by a frame count. The Searcher core only maintains a single bit count to identify frames, allowing settings which change on a frame boundary to be specified as applying to even or odd frames.

The control count block is also responsible for synchronizing the block to frame boundaries. When the block is initialized, it asserts the signal `A_SINTERRUPT` (see "Antenna Interface Timing" on page 22 for details).

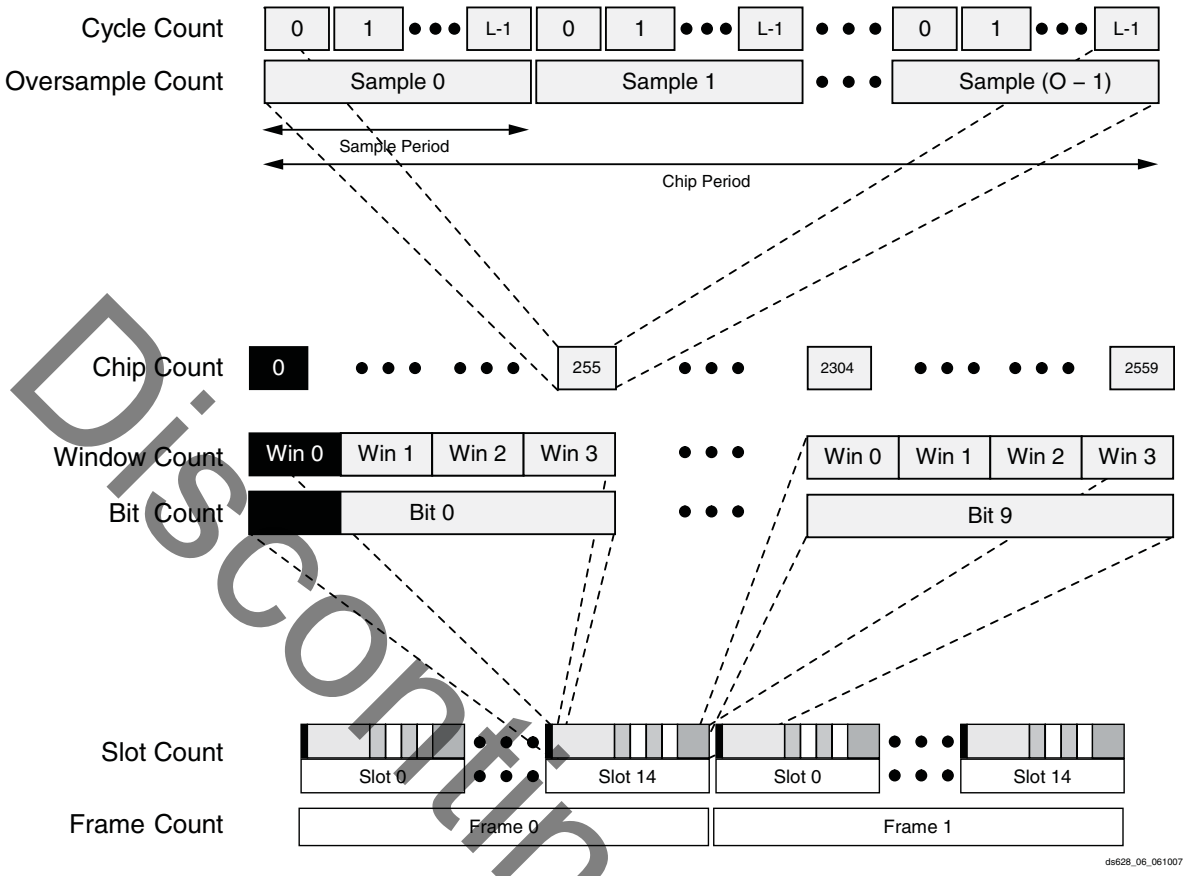


Figure 6: **Control Count Hierarchy**

Scramble Code Advance Block

The scramble code advance block is the first stage in the search processing pipeline. The search configuration written to the block includes details for each search of the scrambling code, and search offset.

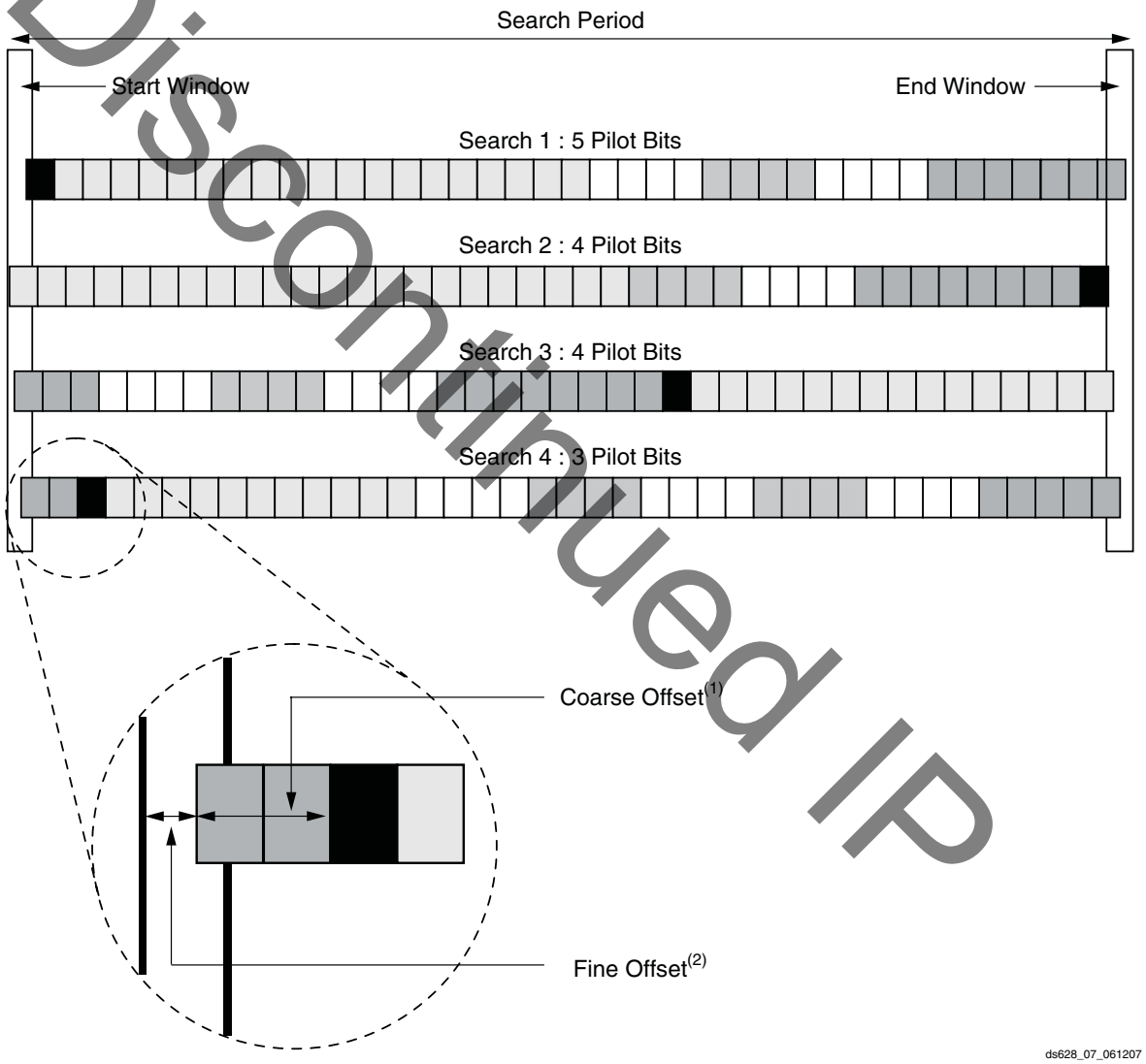
Scramble code generation is defined by Gold Sequences generated using 2-degree, 25-term polynomials called the X and Y polynomials (for details, refer to section 4.3.2.2 of the 3GPP specification TS25.213 *V6.5.0 Spreading and Modulation (FDD) (Release 6)*). Both X and Y sequences are generated using linear feedback shift registers (LFSRs). The lower 24-bits of the X LFSR is set to the specified scrambling code at the start of a frame, with the top bit set to 1. The Y LFSR is set to all 1s at the start of a frame. Each UE has a different offset, since each UE transmission can be offset relative to the global frame count, and each UE has different transmission path delay characteristics.

The scramble code advance block is responsible for generating an initial state for scramble code generation at the start of a search period, taking into account this offset. The block advances the scrambling code by a certain number of windows (known as coarse offset). The remaining fine offset is handled by the scramble code and pilot generation block.

Scramble Code and Pilot Generation Block

At the start of a search period, the initial scramble code generation state is loaded into the scramble code and pilot generation block. This block generates the data sequence that the correlator block correlates against. The block delays the starts of the sequence to account for the fine offset of the search (Figure 7).

Figure 7 uses the same shading as previous diagrams to identify the bits within the DPCCH slot. The scrambling code sequences for each search are not aligned as they have differing offsets and number of pilot bits. The coarse offset of each search can be seen from where the first window in a slot occurs (marked in black). The fine offset of each search is apparent from the start point of the sequence within the start window. The fine offset is required since the correlator block only correlates against complete windows.



Notes:

1. Offsets by greater than one window by shifting start point of correlation sequence in frame.
2. Offset less than one window is implemented by delaying start of correlation within start window.

Figure 7: **Scramble Code Sequences**

Correlator and Coherent Accumulation Blocks

These blocks evaluate the correlation between the scramble code and the sample data stream. The correlation can be expressed as:

$$X(d) = \sum_{w=w_0}^{w=w_0+N-1} \sum_{i=0}^{i=L-1} C_{wL+i} S_{wL+i+d}$$

Equation 1

where:

C_t is the complex scrambling code at time t

S_t is the complex sample at time t

L is the length of the window

$X(d)$ is the coherent correlation result for search over windows starting at w_0

N is the coherent length of the correlation in windows

The correlator block evaluates the inner sum of the equation above, generating a set of results for each window. The coherent accumulation block evaluates the outer sum, accumulating results over a known data sequence, as indicated by the shaded areas in Figure 7 on page 9.

Non-Coherent Accumulation

Eq. 2 expresses the results from the coherent correlation of Eq. 1 in terms of the average real and imaginary values per window of the correlation:

$$NI(d) = Re(X(d))$$
$$NQ(d) = Im(X(d))$$

Equation 2

where

$I(d)$ is the real component of the average correlation result per window

$Q(d)$ is the imaginary component of the average correlation result per window.

To accumulate over a longer time, the core needs to filter or accumulate results non-coherently using the power of the coherent correlation results.

Let $P(d)$ be the power of the average correlation result per window:

$$P(d) = [I(d)]^2 + [Q(d)]^2 = \frac{|X(d)|^2}{N^2}$$

Equation 3

To accumulate the results non-coherently:

$$\bar{P}'(d) = \bar{P}(d) + \alpha NP(d)$$

Equation 4

where:

$\bar{P}(d)$ is the current power delay profile (PDP)

$\bar{P}'(d)$ is the new value of the PDP

α is a scaling value

The value of *NP(d)* is calculated in Eq. 4 from the power of the actual correlation results by dividing Eq. 3 by *N*:

$$NP(d) = \frac{[NI(d)]^2 + [NQ(d)]^2}{N} = \frac{|X(d)|^2}{N}$$

Equation 5

Eq. 4 provides the basic equation for accumulating coherent results of different lengths. The scaling factor α can be chosen such that the accumulation does not overflow when summing over the whole length of a search.

In a real radio environment, different paths suffer from fading. Often a time-weighted average is required for the power delay profile to prevent fading from causing a peak to disappear.

The 3GPP Searcher core uses the following filtering equation to change the PDP by the weighted difference of the current coherent result and the current value of the PDP:

$$\bar{P}'(d) = \bar{P}(d) + \alpha N[P(d) - \bar{P}(d)]$$

Equation 6

Eq. 6 is similar to Eq. 4 used for unfiltered non-coherent accumulation. The Searcher core implements a non-coherent accumulation mode by zeroing the $-\bar{P}(d)$ term in Eq. 6, which gives Eq. 4.

Note: The accumulation mode is also used for initializing the time-weight filtering mode, which can otherwise take many iterations to approach the actual channel conditions if a small value of α is used.

Figure 8 shows a simplified view of the datapath in the non-coherent accumulation block, implementing the operations described in Eq. 6. The circuit within the non-coherent accumulation and filtering block is implemented using signed 18-bit numbers as these match well to the DSP48 elements within the FPGA fabric. The 37-bit numbers generated as the result of the multiply-accumulate operations are scaled back to 18-bit numbers by the shift-right blocks. These blocks also deal with saturation in cases where the multiply-accumulate outputs exceed their expected range.

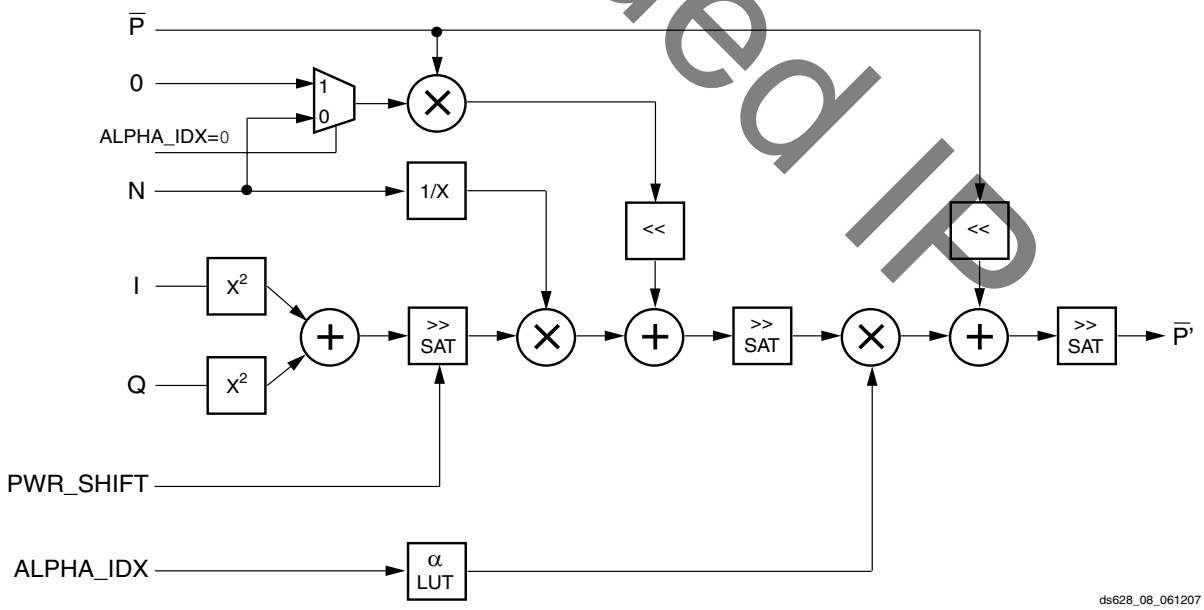


Figure 8: **Non-Coherent Accumulation and Filtering**

The values of PWR_SHIFT and ALPHA_IDX are fields in the search configuration written to the 3GPP Searcher core. The PWR_SHIFT field allows the gain of the search to be adjusted. A PWR_SHIFT of 0 gives a full-range output only if the core is receiving a full-range input signal that matched perfectly to the expected scrambling code. In practice, the power of each UE is a small fraction of the full-scale, and the PWR_SHIFT is used to adjust for this case.

Note: Using PWR_SHIFT can lead to saturation in the data path. This condition is indicate on a per PDP basis by a saturation flag in the signal `SR_SDATAINFO`. This flag is cleared using the CLEAR_SAT field in the search configuration.

The ALPHA_IDX field specifies a number of possible α values for the time-weighted average. This index is converted to the actual α value used by a small lookup table.

Refer to "Search Configuration Register 0" on page 18 for more details on PWR_SHIFT and ALPHA_IDX and CLEAR_SAT fields.

Port Descriptions

The 3GPP Searcher core is designed to be used as co-processor to a general purpose processor or DSP. OCP-compatible interfaces are used to provide a consistent interface adaptable to many system bus types. Refer to "Register and Memory Maps" on page 17 for details of data transfer across the OCP data and address signals.

Port Diagrams

Figure 9 shows the top-level interface to the core.

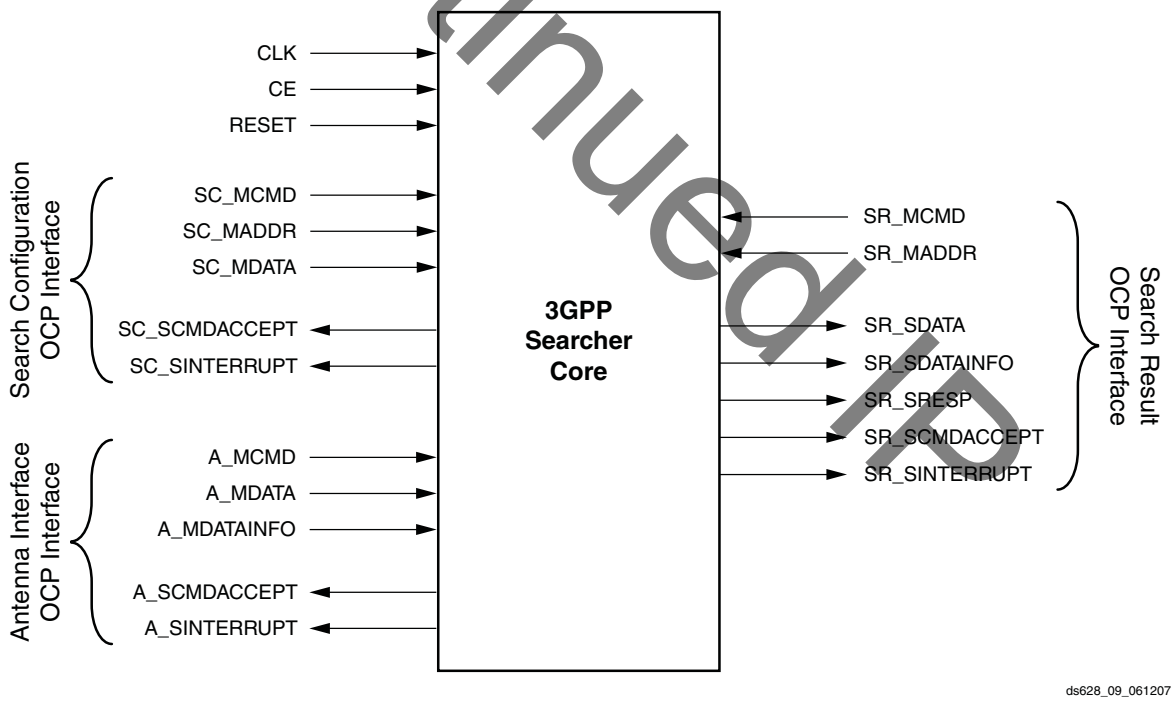


Figure 9: **3GPP Searcher Core Ports**

Port Descriptions and Definitions

Clock and Reset

Table 1 lists the clocks and resets for the Searcher core.

Table 1: **Clock and Reset(1)**

Port Name	I/O	Width	Description
CLK	I	1	**Chip Rate Processing Clock**. Used to synchronize all OCP-compatible interfaces.
CE	I	1	**Clock Enable (optional).** Clock enable halts all internal clocks when asserted.(2)
RESET	I	1	**Active-High Synchronous Reset.(3)**

Notes:
1. Clock and reset are common to all blocks.
2. CE is not defined in OCP specification. Asserting CE during OCP accesses could result in the block not complying with OCP specification.
3. OCP reset is specified as active Low. Active-High reset is adopted to be compatible with other LogiCORE modules. An inverter should be placed before reset if using OCP reset signal.

Search Configuration Interface

Search configuration OCP interface port definitions are listed in Table 2.

Table 2: **Search Configuration OCP Interface Ports(1)**

Port Name	I/O	Width	Description
SC_MCMD	I	3	**OCP Master Command** only supports the following commands: `0xb000`: Idle `0xb001`: Write
SC_ADDR	I	4	**OCP Master Address.** Address of search configuration register to be written.(2)
SC_MDATA	I	32	**OCP Master Data.** Data to be written to search configuration register.(2)
SC_SCMDACCEPT	O	1	**OCP Slave Command Accept** indicates the slave has accepted the command from the master.
SC_SINTERRUPT	O	2	**OCP Slave Interrupt:** Bit 0: Indicates that the master should write a new block of searches. Cleared when first register of the search configuration is written. Bit 1: Indicates that core is ready for another configuration to be written.

Notes:
1. OCP Interface for writing search configuration.
2. See "Antenna Interface Register Map" on page 21.

Antenna Data Interface

Table 3 describes the antenna data OCP interface ports.

Table 3: **Antenna Data OCP Interface Ports(1)**

Port Name	I/O	Width	Description
A_MCMD	I	1	**OCP Master Command.** Only supports the following commands: `0xb000`: idle `0xb001`: write
A_MDATA	I	16	**Antenna Data.** I and Q sample data for the antenna.(2)
A_MDATAINFO	I	1	**OCP Master Data Info.** Bit 0 is set when sample data is synchronized to external timing reference. All offsets in the search configuration are relative to this reference. Only set for data on antenna 0.
A_SCMDACCEPT	O	1	**OCP Slave Command Accept** indicates the slave has accepted the command from the master.
A_SINTERRUPT	O	1	**OCP Slave Interrupt** indicates that the slave is expecting a sample synchronized to external timing reference. Cleared when data is written with `A_MDATAINFO[0]` set.

Notes:
1. Antenna data is time-interleaved on the interface.
2. See "Antenna Interface Register Map" on page 21.

Search Results Interface

Table 4 defines the search results OCP interface ports.

Table 4: **Search Results OCP Interface Ports**

Port Name	I/O	Width	Description
SR_MCMD	I	3	**OCP Master Command.** Valid Commands are: `0b000` : idle `0b010` : read
SR_MADDR	I	23	**OCP Master Address.** Address specifying PDP and delay offset in PDP to be read.(1)
SR_SDATA	O	18	**OCP Data.** PDP result data.(1)
SR_SDATAINFO	O	1	**OCP Data Information.** Bit 0: Indicates that saturation has occurred for this PDP. Cleared using the CLEAR_SAT field in search configuration register 0.
SR_SRESP	O	2	**OCP Slave Response.** Valid values are: `0b00` : null response `0b01` : data valid response
SR_SCMDACCEPT	O	1	**OCP Command Accept.**

Table 4: **Search Results OCP Interface Ports**

Port Name	I/O	Width	Description
SR_SINTERRUPT	O	1	**OCP Slave Interrupt** indicates completion of a block of search, and that results can be read from the results memory. Cleared by reading memory.

Notes:
1. See "Search Result Memory Map" on page 21.

Core Generation and Customization

The 3GPP Searcher core is generated and parameterized through the customize option of CORE Generator™ software. The customization GUI also provides feedback about the complexity of the design through a number of design metrics displayed on the second page of the GUI.

GUI Parameters

Core parameters that can be set via the CORE Generator GUI are listed in Table 5.

Table 5: **GUI Parameters**

Parameter	Values	Description
Clock Enable	True, False	True = component has CE pin False = component does not have CE pin
Oversample_Rate	1, 2, 4, 8	Number of samples per chip in antenna data stream(1).
Clock_Rate	8 … 256	Number of clock cycles available to process one chip. Clock rate must be a multiple of the oversample rate(2).
Antennae	1 ...16	Number of antennae. Must be smaller or equal to number of clocks per sample (clock rate divided by oversample rate).
Quantization	4 … 8	Number of bits used to represent samples.
PDP_Length	4, 8, 16, 32, 64,128, 256	Length of PDP generated by a search expressed in chips. Same as the basic window used to accumulate the correlation results.
Number_of_Searches	1 … 256	Number of searches that are can be calculated concurrently during a search period.
Number_of_Results	1 … 1024	Number of result PDPs stored in the search results memory.
Search_Length	2560 … 38400	Length of search period expressed in chips. Must be a multiple of the PDP Length.(3)

Notes:
1. Oversample_Rate corresponds to O in Figure 6 on page 8.
2. Clock_Rate corresponds to the product of parameters O and L in Figure 6 on page 8
3. This parameter determines the search period as shown in Figure 7 on page 9.

Design Metrics

These design metrics (Table 6) presented on the second page of the CORE Generator GUI provide feedback to the designer on the complexity of the core in terms of the memory required and processing operations performed per unit time, effecting the size of the final core.

Table 6: **Design Metrics**

Design Metric	Description
Number of Correlation Points	Measures how many correlation points are calculated concurrently during a search period. Determines the size of memories used to store the coherent correlation results.
Number of MACs	Number of multiply-accumulate elements in correlator block. Determines the basic size of correlator block.
Number of Sample SRL16s	Number of sample SRL16s used to construct sample shift register in the correlator block.
Number of Accumulation Chains.	Number of parallel accumulation chains used. The correlator block can adapt to deal with cases where the number of searches performed concurrently is not a factor of the number of clock cycles available per sample, minimizing the number of MACs required. To achieve this adaptation, the correlator has a number of parallel accumulation chains which may represent a significant overhead when the number of chains is a significant proportion of the number of MACs in a design. Also, when the number of accumulation chains is greater than one, restrictions may be placed on the number of antennae supported.

Register and Memory Maps

Searcher Core Operational Flow

The following sections define the register and memory maps for the three OCP-compatible interfaces in the 3GPP Searcher core. Figure 10 shows the typical operational flow of these registers. The operation of the 3GPP Searcher core is pipelined based on search periods. The length of a search period is determined by the Search_Length parameter of the core.

In the first search period, the block of search configurations is written to the Searcher core via the search configuration registers ("Write Config 0" in Figure 10). The search configuration contains details of the search including the result ID of the PDP where the result is stored, or updated. At the start of the next search period, the Searcher core starts searching on the new block of configurations ("Search on Config 0" in Figure 10). The core then correlates against data written on the antenna interface. After the end of the search, the PDP results can be read via the search result memory ("Read Results for Config 0" in Figure 10).

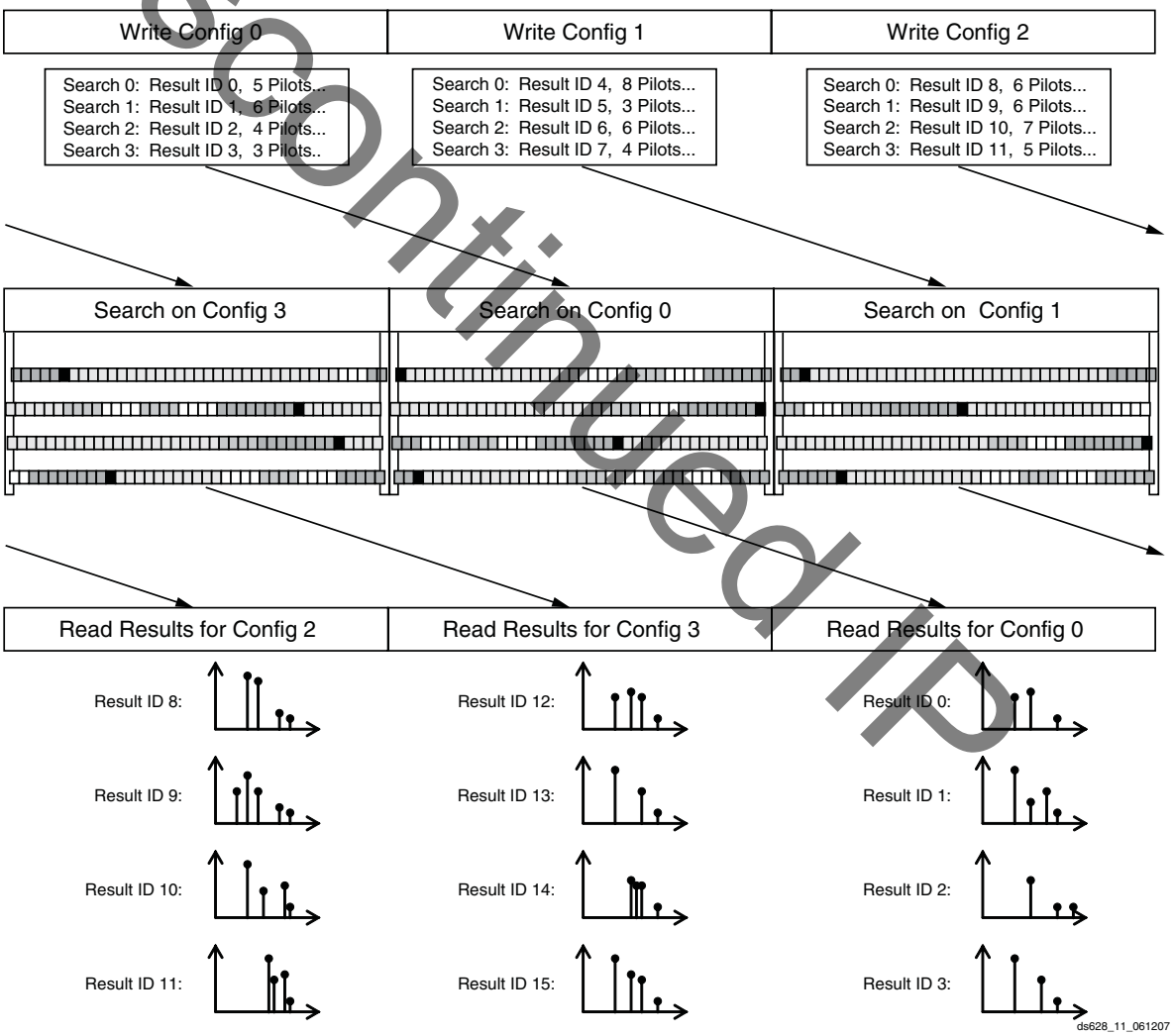


Figure 10: **Searcher Core Pipelined Operation**

Search Configuration Interface

Search Configuration Register Map

Table 7 defines the search configuration register map.

Table 7: **Search Configuration Register Map (SC_MADDR)**

Address	Description
0x00	**Search Configuration Register 0** contains configuration for the results processing.
0x04	**Search Configuration Register 1** configures the delay offset and other aspects of search.
0x08	**Search Configuration Register 2** configures the scrambling code and the antenna to be searched on.
0x12	**Search Configuration Register 3**. Writing to this register causes the search configuration to be transferred to the scramble advance unit. Further writes to the search configuration registers are ignored until the scramble advance unit completes advancing the scrambling code.(1)

Note:
1. See "Search Results Interface" on page 26 for timing details.

Search Configuration Register 0

Table 8 defines the search configuration register 0, used for configuring PDP results processing.

Table 8: **Search Configuration Register 0 (SC_MADDR = 0x00)**

Range	Field	Description
31:20	RSVD	**Reserved**. Set to 0.
19	ENABLE	**Search Enable.** 0x0 : Search Disabled. 0x1 : Search Enabled.
18	CLEAR_SAT	**Clear Saturation Flag** 0 : Do not clear Saturation flag. 1 : Clear Saturation flag. This bit is used to clear the saturation flag for the PDP with the specified result index. The saturation flag is reported on the SR_SDATAINFO[0] pin. The bit is cleared when the result PDP is updated. Used in conjunction with the PWR_SHIFT field which adjusts the power gain on a particular search.
17:14	PWR_SHIFT[3:0]	**Power Shift.** Shifts power results by specified amount. Power is saturated if power shift overflows.

Table 8: **Search Configuration Register 0 (SC_MADDR = 0x00) *(Continued)***

Range	Field	Description
13:10	ALPHA[3:0]	**Filter Mode and Time Weighting.** Used to select the filter mode and time weighting. When set to 0x0, the filter works in instantaneous mode (see Eq. 4). This mode gives a result for the current search period. The coherent results are accumulated with no time weighting to give an average correlation result for a single search period. This mode can also be used to initialize the result for a time-weighted filter. For other values of α, the filter acts in time-weighted mode (see Eq. 6). The weighting of each coherent results to the filtered result is specified by the value of α as follows.(1) 0x0: Instantaneous Mode 0x1: 1/N 0x2: 2/3N 0x3: 1/2N 0x4: 1/3N 0x5: 1/4N 0x6: 1/6N 0x7: 1/8N 0x8: 1/12N 0x9: 1/16N 0xA: 1/24N 0xB: 1/32N 0xC: 1/48N 0xD: 1/64N 0xE: 1/96N 0xF: 1/128N
9:0	RESULT_ID[9:0]	**Result Identifier.** Identifier of the PDP to be updated by search. Number of valid results is set by the Number_of_Results parameter.

Search Configuration Register 1

Table 9 defines the search configuration register 1, used to configure search offset and number of pilot bits.

Table 9: **Search Configuration Register 1 (SC_MADDR = 0x04)**

Range	Field	Description
31:22	RSVD	**Reserved.** Set to 0.
21:19	NPILOT1[2:0]	**Number of Pilot Bits in Odd Frames** determines the number of pilot bits in odd numbered frames that are coherently correlated against. The Searcher core automatically swaps to this value if a frame boundary is crossed during a search: 0x0 : reserved 0x1 : reserved 0x2 : 3 pilots 0x3 : 4 pilots 0x4 : 5 pilots 0x5 : 6 pilots 0x6 : 7 pilots 0x7 : 8 pilots

Table 9: **Search Configuration Register 1 (SC_MADDR = 0x04) *(Continued)***

Range	Field	Description
18:16	NPILOT0[2:0]	**Number of Pilot Bits in Even Frames.** Determines the number of pilot bits in even numbered frames that are coherently correlated against. The Searcher core automatically swaps to this value if a frame boundary is crossed during a search: 0x0 : reserved 0x1 : reserved 0x2 : 3 pilots 0x3 : 4 pilots 0x4 : 5 pilots 0x5 : 6 pilots 0x6 : 7 pilots 0x7 : 8 pilots
15:12	DLY_SLOT[3:0]	**Slot Delay Offset.** Valid values: 0 to 14.
11:0	DLY_CHIP[11:0]	**Chip Delay offset.** Valid values: 0 to 2559.

Search Configuration Register 2

Table 10 defines the search configuration register 2. This register configures the scrambling code and antenna.

Table 10: **Search Configuration Register 2 (SC_MADDR = 0x08)**

Range	Field	Description
31:30	RSVD	**Reserved.** Set to 0.
29:24	ANTENNA[5:0]	**Antenna to Run Search On**. Valid values are 0 to number of antenna minus 1.
23:0	SCR0[23:0]	**Initial Value of X LFSR Register** to generate scramble code.

Search Configuration Register 3

Table 11 defines the search configuration register 3. Writing to this register causes the search configuration to be transferred to the scramble advance unit. The register is a dummy register with no contents, and its only effect is to start the scramble advance unit. While the scramble advance unit is operating, further register writes are ignored (see "Search Configuration Interface Timing" on page 22“).

Table 11: **Search Configuration Register 3 (SC_MADDR = 0x12)**

Range	Field	Description
31:0	RSVD	**Reserved.** Set to 0.

Antenna Interface Register Map

Antenna Data

Table 12 defines the antenna data fields.

Table 12: **Antenna Data Fields (A_MDATA[15:0])**

Range	Field	Description
15:8	QDATA	**Q Component of Sample** (2's complement number). Valid range is determined by Quantization core parameter.
7:0	IDATA	**I Component of Sample** (2's complement number). Valid range is determined by Quantization core parameter.

Search Result Memory Map

The result PDPs are updated when a search is being run, every time the end of a coherent correlation occurs. For searches using the time-weighted filtering (ALPHA_IDX field not equal to 0), then the results will be valid through-out a search. For searches using the accumulation mode (no filtering), then the results are only valid at the end of a search. In this case, if performing a search on the same user in the next search period, then the search should be configured to use a different result ID, so that the previous search result can be read safely.

Search Result Memory Map

Table 13: **Search Results Address: SR_MADDR[22:0](1)**

Range	Field	Width	Description
22:13	RESULT	10	**Result Identifier** Identifies PDP. Valid values are: 0 to *Number_of_Results* – 1.
12:2	DLY	11	**Delay Offset of result.** Valid values are: 0 to *PDP Length* × *Oversample Rate* – 1.
1:0	Word Alignment	2	**OCP Word Alignment.** Bits set to 0.

Note:
1. Address space is configured for maximum window size and maximum number of results. Unused bits for smaller designs should are set to 0.

Search Result Data

Table 14: **Search Result Data: SR_MDATA[17:0](1)**

Range	Field	Description
17:0	PWR	Power

Note:
1. Returns power for PDP and offset specified in address.

Timing Diagrams

The 3GPP Searcher core uses OCP-compatible v2.0 interfaces for each of the main interfaces, allowing the interfaces to be easily adapted to a variety of bus protocols.

Search Configuration Interface Timing

The search configuration interface is a write-only OCP-compatible interface used to write the search configuration to the 3GPP Searcher core. Figure 11 shows the timing when writing a block of search configurations to the core. The transfer is initiated by the core asserting `SC_SINTERRUPT[0]`, indicating the core is ready for a new block of search configurations to be written. The interrupt is cleared when the first configuration word is written.

The search configuration registers are written with configuration data as specified in the register map given in "Antenna Interface Register Map" on page 21. The final register written is search configuration register 3, causing the search configuration to be transferred to the scramble advance block. While the scramble advance block is advancing the scramble code, further data cannot be written.

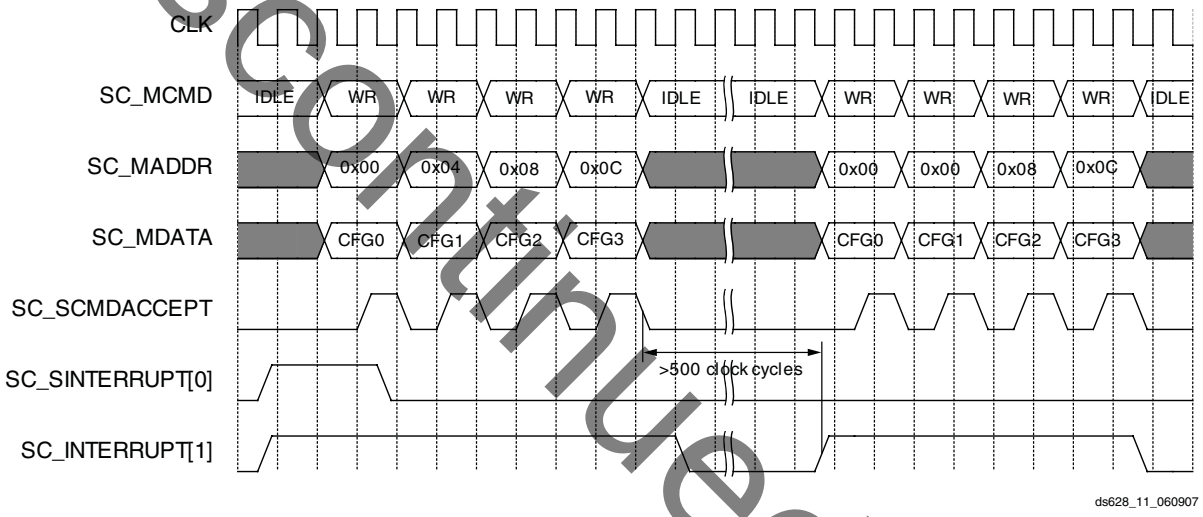


Figure 11: **Search Configuration Timing**

The second interrupt flag `SC_SINTERRUPT[1]` is used to indicate when further data can be written to the search configuration. This interrupt can also be used as a DMA request signal. When a complete block of search configuration is written, the DMA request is not re-asserted. The request is only re-asserted when the next search period starts, and a further block of search configurations can be written.

Antenna Interface Timing

The antenna interface is a write-only OCP-compatible interface. As shown in Figure 12, antenna data is written as a block starting with the lowest numbered antenna, with the transfer repeating every sample period. The signal `A_SCMDACCEPT` indicates when the antenna data is expected.

The timing diagram in Figure 12 is dependent on a number of the core parameters. The number of antenna specifies the length of the burst transferred. The sample period is determined by the core's clock rate divided by the oversample rate. In this example, the number of antennae is 4, the clock rate is 20, and the oversample rate is 2, giving a sample period of 10 clock cycles.

The antenna interface is synchronized to the internal processing counts of the core (shown in Figure 6 on page 8). The data for antenna 0 is transferred when the cycle count is 0, antenna 1 when cycle count is 1, etc., up to the number of antennae.

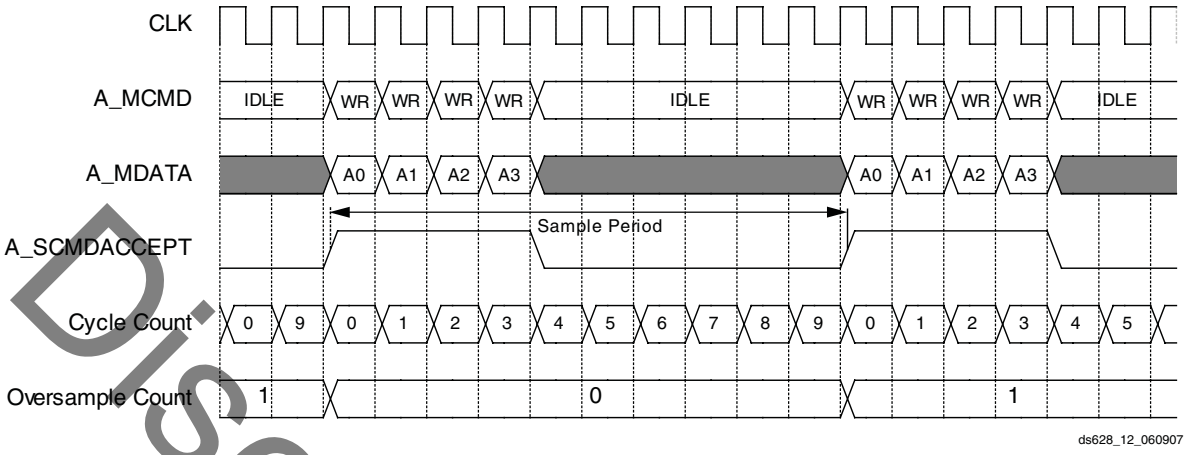


Figure 12: **Antenna Interface Basic Data Transfer**

The signal A_SCMDACCEPT signal forces the master data transfers to synchronize to the cycle count and the internally processing rate of the Searcher core (Figure 13). If the master writes data to the core early, then A_SCMDACCEPT is not asserted until the cycle count reaches the correct value, thus throttling the master data transfer rate.

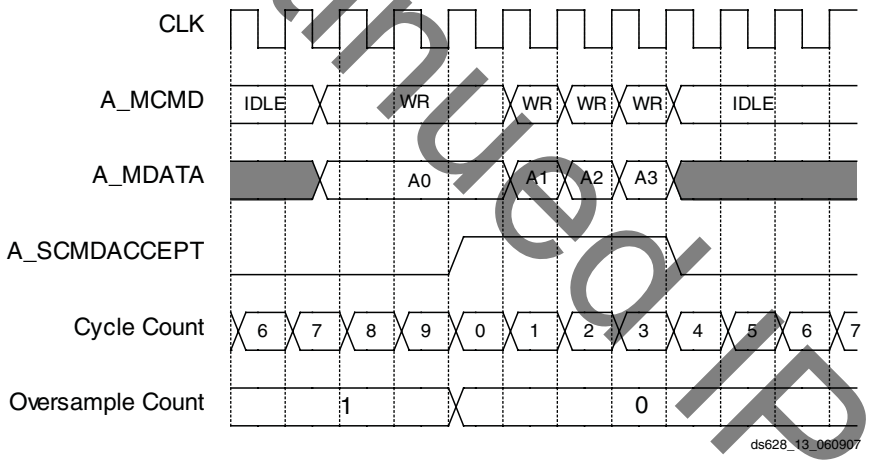


Figure 13: **Core Throttling Master Transfer Rate**

If the master does not supply antenna data on the expected cycle, then the cycle count is not incremented and processing in the core is stalled (Figure 14). Since processing in the core stalls, additional clock cycles are required to meet the real-time processing requirements of the core.

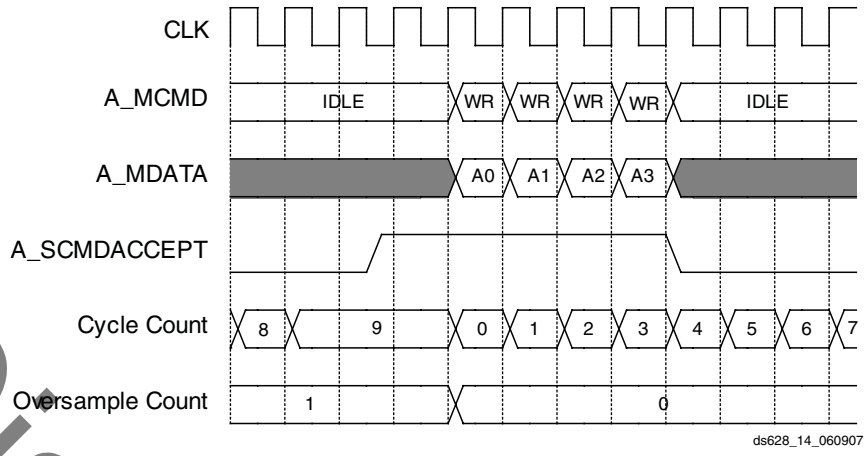


Figure 14: **Master Throttling Core Processing Rate**

Figure 15 shows the synchronization of the core to an external framing reference. The signal A_MDATAINFO is asserted when writing the first sample of antenna zero when frame synchronization occurs. The signal A_SINTERRUPT indicates that the core is expecting the frame synchronization. A_MDATAINFO, and A_SINTERRUPT occur at the same time when the core is synchronized. The offsets specified in the search configuration are used to offset the searches from this reference point.

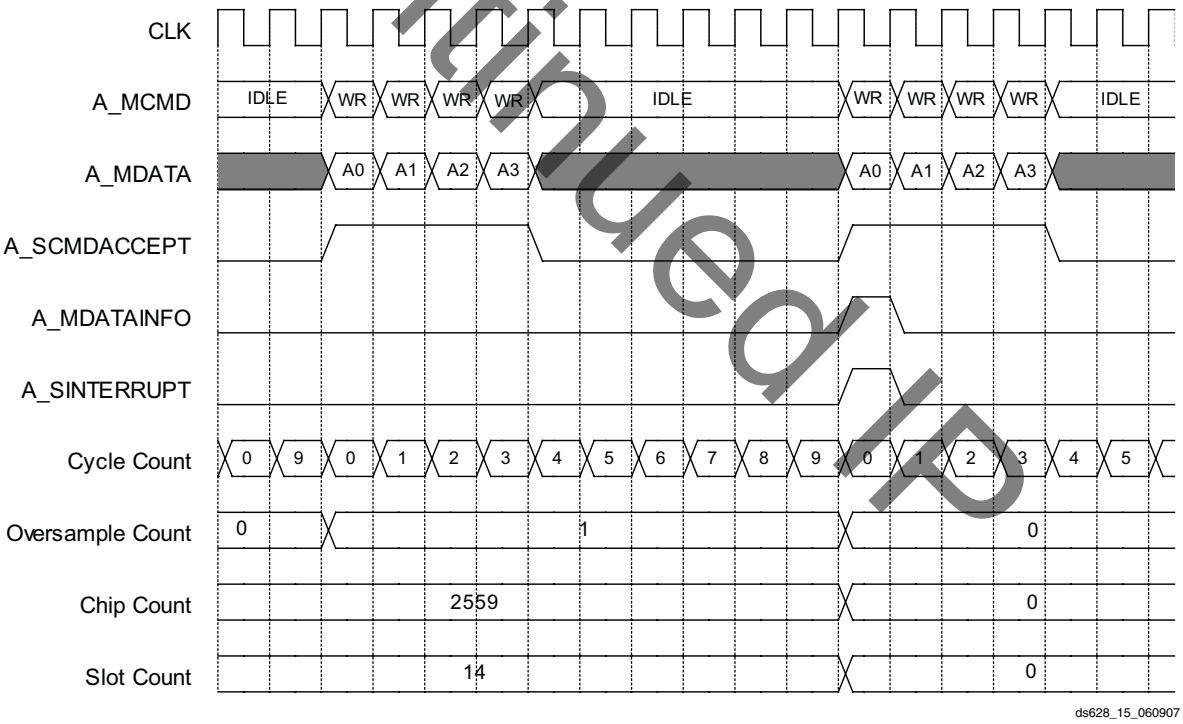


Figure 15: **Antenna Data Frame Synchronization**

Figure 16 shows how the core's initial synchronization to the frame sync signal on A_MDATAINFO. The core asserts A_SINTERRUPT when expecting the signal A_MDATAINFO and then stalls until the A_MDATAINFO is written. While the core is stalled, the cycle count does not increment, and A_SCMDACCEPT is continuously asserted since the core is expected data to be written with A_MDATAINFO set. Data written to the core while waiting for A_MDATAINFO is ignored. When the A_MDATAINFO is written, then the core starts processing again, in sync with the incoming antenna data stream.

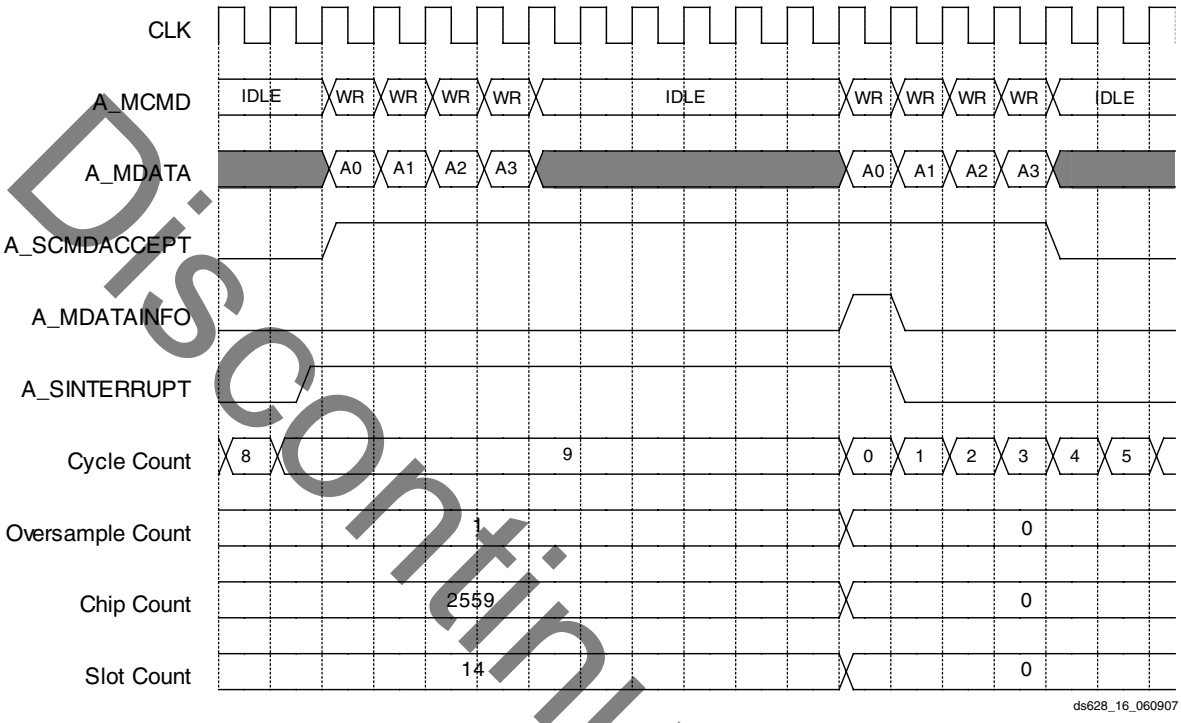


Figure 16: **Initial Synchronization**

Search Results Interface

The search result interface is a pipelined read-only OCP-compatible interface. The signal SR_SINTERRUPT indicates when a current block of searches is complete, and the results can be read (Figure 17). The results can be read. The first read from the memory clears the interrupt.

The results are read for each PDP using the address specified in "Register and Memory Maps" on page 17. After issuing the OCP read command, there is a three-clock latency from the SR_SCMDACCEPT assertion to data valid as indicated by the signal SR_SRESP.

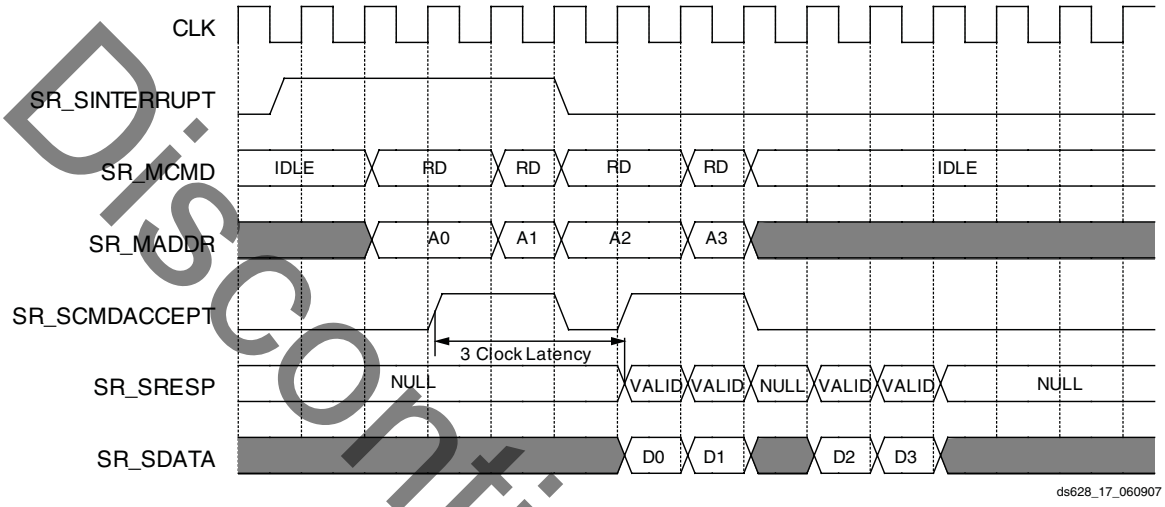


Figure 17: **Search Results Timing**

Performance Characteristics

Table 15 through Table 17 show the performance of the 3GPP Searcher core in terms of resource usage and maximum achieved operating frequency for different FPGA families.

Note: These results were obtained with ISE™ v9.2.

The resource count and speed of the core can change depending on the surrounding circuitry of the user design. Therefore, these figures are only a guideline.

The tool settings used to achieve these results are as follows:

```
map -c 1 -ol high
par -ol high
```

Note: The tool settings can have a significant effect on area use and speed. The Xplorer script can be used to find the optimal settings.

Table 15: **Spartan-3A DSP Resource Utilization**

XCO Parameter	Case 1 – Femto Cell	Case 2 – Pico Cell	Case 3 – Macro Cell
Clock_Enable	False	False	False
Oversample_Rate	2	2	2
Clock_Rate	32	32	32
Antennae	2	4	8
Quantization	5	6	7
Number of Searches	8	16	16
Number of Results	16	32	64
PDP_Length	8	16	64
Search_Length	2560	2560	2560
Utilization			
Xilinx Device	XC3SD1800A	XC3SD1800A	XC3SD1800A
Slices(1)	1195	1781	5026
LUTs	1781	2659	7943
FFs	1808	2704	7035
Block RAMs (18k)	3	3	13
DSP Blocks	5	5	5
Maximum Clock Frequency(2)	170	172	153

Notes:
1. Area and maximum clock frequencies are provided as a guide and can vary with new releases of the Xilinx implementation tools.
2. Maximum clock frequencies are shown in MHz for -4 parts. Clock frequency does not take jitter into account and should be derated by an amount appropriate to the clock-source jitter specification.

Table 16: **Virtex-4 Resource Utilization**

XCO Parameter	Case 1 – Femto Cell	Case 2 – Pico Cell	Case 3 – Macro Cell
Clock_Enable	False	False	False
Oversample_Rate	2	2	2
Clock_Rate64	64	64	64
Antennae	2	4	8
Quantization	5	6	7
Number of Searches	8	16	32
Number of Results	16	32	64
PDP_Length	8	16	32
Search_Length	2560	2560	2560
Utilization			
Xilinx Device	XC4VLX25	XC4VLX25	XC4VLX25
Slices(1)	1028	1348	2769
LUTs	1696	2283	5195
FFs	1691	2200	4245
Block RAMs (18k)	3	3	10
DSP Blocks	5	5	5
Maximum Clock Frequency(2)	270/339	263/329	249/301

Notes:

1. Area and maximum clock frequencies are provided as a guide. They may vary with new releases of the Xilinx implementation tools.
2. Maximum clock frequencies are shown in MHz for -10/-12 parts. Clock frequency does not take jitter into account and should be derated by an amount appropriate to the clock-source jitter specification.

Table 17: **Virtex-5 Resource Utilization**

XCO Parameter	Case 1 – Femto Cell	Case 2 – Pico Cell	Case 3 – Macro Cell
Clock Enable	False	False	False
Oversample_Rate	2	2	2
Clock_Rate	64	64	64
Antennae	2	4	8
Quantization	5	6	7
Number of Searches	8	16	32
Number of Results	16	32	64
PDP_Length	8	16	32
Search_Length	2560	2560	2560
Utilization			
Xilinx Device	XC5VLX30	XC5VLX30	XC5VLX30
LUT/FF Pairs(1)	1814	2348	4414
LUTs	926	1262	2834
FFs	1688	2197	4180
Total Block RAMs(3)			
Block RAMs (36k)	0	0	2
Block RAMs (18k)	3	3	6
DSP Blocks	5	5	5
Maximum Clock Frequency(1,2)	287/378	276/364	260/332

Notes:

1. Area and maximum clock frequencies are provided as a guide. They may vary with new releases of the Xilinx implementation tools.
2. Maximum clock frequencies are shown in MHz for -1/-3 parts. Clock frequency does not take jitter into account and should be derated by an amount appropriate to the clock-source jitter specification.
3. Represents the total number of 36k block RAMs used when map is run. In reality, two 18k block RAM primitives can usually be packed together, giving an absolute minimum total block RAM usage of block RAMs (36k) + (block RAMs (18k) /2) (rounded up).

Support

Xilinx provides technical support for this LogiCORE product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled *DO NOT MODIFY*.

Ordering Information

The 3GPP Searcher core is provided under the SignOnce IP Site License and can be generated using the Xilinx CORE Generator software v9.2i or higher. The CORE Generator software is shipped with Xilinx ISE™ Foundation™ Series Development software.

Once purchased, the core may be downloaded from the Xilinx IP Center for use with the Xilinx CORE Generator v9.2i and higher. The Xilinx CORE Generator software is bundled with the ISE Foundation software at no additional charge.

Contact your local Xilinx sales representative for pricing and availability of additional Xilinx LogiCORE modules and software. Information about additional Xilinx LogiCORE modules is available on the Xilinx IP Center.

Revision History

Date	Version	Revision
08/08/07	1.1	Initial Xilinx release