**Multiplexing** is used in the cases where the signals of lower bandwidth and the transmitting media is having higher bandwidth. In this case, the possibility of sending a number of signals is more. In this the signals are combined into one and are sent over a link which has greater bandwidth of media than the communicating nodes.

**Time-Division Multiplexing TDM** : Many signals are assembled into one signal using a multiplexer. Then they are reassembled with respect to time at the receiving end using a demultiplexer.

Multiplexer assembles the data which is then reassembled using a Demultiplexer.

Thus the number of signals to be sent with time could be controlled using this method. It is used to control internet traffic.

Also can be used if the signal needs to be transmitted slowly compared to speed in which it is processed.

Examples of utilizing TDM include digitally transmitting several telephone conversations over the same four-wire copper cable

**Synchronous TDM** : We need to transmit data. It provides a certain amount of time slots to all users. Thus guarantees the users data will be transmitted within the time period. However if some user had no data to send. Its time slot gets wasted. As the users are fixed. Their rate of data transmission is same. That is why synchronous.

**Asynchronous TDM**:It provides time slots to only active users. Thus no guarantee the users data will when be transmitted. However it saves any wastage of time slots. If the devices have nothing to transmit, then their time slot is allocated to another device.

**Frequency-Division Multiplexing** : Used to perform same task as TDM. Many signals are coupled with signals of different frequencies within some specified bandwidth.. Then they are assembled into one signal using a multiplexer. Say we have a bandgap of 2000 KHz and we need to keep signals 2KHz apart. Then we can transmit 1000 signals at one time.At the receiver, the signals are separated using band-pass filters to isolate the individual frequency channels and along with it de-assembled using a demultiplexer.

*There is a suitable frequency gap between the 2 adjacent signals to avoid over-lapping. Since the signals are transmitted in allotted time so this decreases the probability of collision. The frequency spectrum is divided into several logical channels, in which every user feels that they possess a particular bandwidth. A number of signals are sent simultaneously on the same time allocating separate frequency band or channel to each signal. It is used in radio and TV transmission. Therefore to avoid interference between two successive channels* ***Guard bands*** *are used.*

FDMA : When FDM is used to allow multiple users to share a single physical communications medium (such as a [coaxial cable](https://en.wikipedia.org/wiki/Coaxial_cable) or [microwave](https://en.wikipedia.org/wiki/Microwave) beam), it is called frequency-division multiple access (FDMA).

**TDM vs FDM** : FDM divides the channel into two or more frequency ranges that do not overlap, while TDM divides certain time periods to each channel . Thus, for TDM, each signal uses all of the bandwidth some of the time, while for FDM, each signal uses a small portion of the bandwidth all of the time. Thus, TDM is more flexible in allocating more time period to signals who need to send more data at that time but FDM cannot have this flexibility as certain bandwidths are allocated for each signal all the time.

Thus, FDM will take less time to transmit a signal, so used in places where time is of utmost priority like real time data sending.

**OFDM(Orthogonal FDM)** in FDM systems carriers are far apart with respect to each other and in OFDM systems carriers are densely packed and are orthogonal to the other carriers. The data is divided into a large number of carriers located at precise frequencies. Orthogonal means peak of one carrier occurs at null of the other. The orthogonality requires that the sub-carrier spacing is Δ f = [Hertz](https://en.wikipedia.org/wiki/Hertz), where *T*U [seconds](https://en.wikipedia.org/wiki/Second) is the useful symbol duration (the receiver-side window size), and *k* is a positive integer, typically equal to 1. Hence OFDM system is bandwidth efficient compare to FDM system. In FDM system carriers are not orthogonal.

**OFDMA** (**Orthogonal frequency-division multiple access)** is a type of OFDM coupled with TDM.[Multiple access](https://en.wikipedia.org/wiki/Multiple_access) is achieved in OFDMA by assigning subsets of [subcarriers](https://en.wikipedia.org/wiki/Carrier_wave) to individual users. This allows simultaneous low-data-rate transmission from several users. IN TDM several users could send information but with some queuing system i.e. there used to be a time lag to send the data. Here it could be send simultaneously.

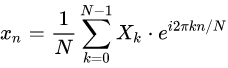
Low-data-rate users can send continuously with low transmission power instead of using a "pulsed" high-power carrier. Constant delay, and shorter delay, can be achieved.

**SC-FDMA**(Single Carrier [frequency-division multiple access](https://en.wikipedia.org/wiki/Frequency-division_multiple_access) ) :

In it various users can transmit info simultaneously. This is divided into the various frequencies and transferred rather than providing separate bandwidths to various users.Thus it will need an additional DFT processing unit before OFDMA.

**FFT (Fast Fourier Transform)**  is a faster way to calculate discrete fourier transform of a sample of signals. Ask me or f=refer to wiki page for more details.

**IFFT (inverse fourier transform)**



**Orthogonal functions** : Functions whose vector product is zero. The vector product will be defined as integral of their multiplication in the real line. This property can be exploited a lot. Ask me for more details. Eg - sin,cos

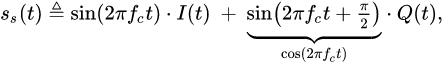
two **signals** are said to be **orthogonal** if they are mutually independent. This in practice means that they do not interfere with each other or that their effects on each other cancel out.

Usually, the subcarriers are finite-length sinusoids of containing N samples of the waveform.Orthogonality of sub-carriers simply means their correlation is zero.

**QAM (quadrature amplitude modulation)** is a method of combining two amplitude-modulated (AM) signals into a single channel, thereby doubling the effective bandwidth. QAM is used with pulse amplitude modulation (PAM) in digital systems, especially in wireless applications.

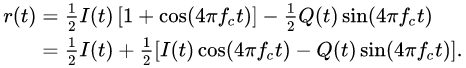
The two carrier waves of the same frequency are [out of phase](https://en.wikipedia.org/wiki/Out_of_phase) with each other by 90°, a condition known as [orthogonality](https://en.wikipedia.org/wiki/Orthogonal) and as [quadrature](https://en.wikipedia.org/wiki/Quadrature_phase). Being the same frequency, the modulated carriers add together, but can be coherently separated (demodulated) because of their orthogonality property.

In a QAM signal, one carrier lags the other by 90°, and its amplitude modulation is customarily referred to as the [in-phase component](https://en.wikipedia.org/wiki/In-phase_and_quadrature_components), denoted by *I*(*t*). The other modulating function is the [quadrature component](https://en.wikipedia.org/wiki/In-phase_and_quadrature_components), *Q*(*t*). So the composite waveform is mathematically modeled as:



where *f*c is the carrier frequency. At the receiver, a [coherent demodulator](https://en.wikipedia.org/wiki/Product_detector) multiplies the received signal separately with both a [cosine](https://en.wikipedia.org/wiki/Cosine) and [sine](https://en.wikipedia.org/wiki/Sine) signal to produce the received estimates of *I*(*t*) and *Q*(*t*). For example:





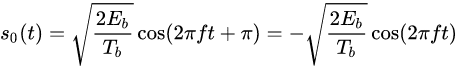
[Low-pass filtering](https://en.wikipedia.org/wiki/Low-pass_filter) *r*(*t*) removes the high frequency terms (containing 4π*f*c*t*), leaving only the *I*(*t*) term. This filtered signal is unaffected by *Q*(*t*), showing that the in-phase component can be received independently of the quadrature component. Similarly, we can multiply *s*c(*t*) by a sine wave and then low-pass filter to extract *Q*(*t*).

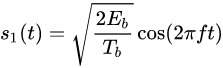
<https://www.electronics-notes.com/articles/radio/modulation/quadrature-amplitude-modulation-what-is-qam-basics.php>

Phase shift keying :

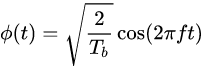
BPSK :( Binary Phase Shift Keying ) : . It uses two phases which are separated by 180° and so can also be termed 2-PSK. As it uses just one phase, it will handle noise more efficiently as other forms of PSKs.However, it will handle only one bit/symbol.

The phases are





Thus the signal can be specified using one basis function

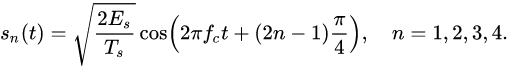


**QPSK** (Quadrature Phase shift keying) :

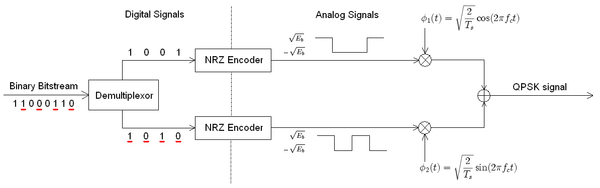
Uses 4 phases separated by 90 deg . So can handle 2 bits/symbol.

Thus QPSK can be used either to double the data rate compared with a BPSK system while maintaining the *same* [bandwidth](https://en.wikipedia.org/wiki/Bandwidth_(signal_processing)) of the signal, or to *maintain the data-rate of BPSK* but halving the bandwidth needed.

Thus, signal can be represented as ,



Thus we will have two basis functions for representing the signal



The binary data stream is split into the in-phase and quadrature-phase components. These are then separately modulated onto two orthogonal basis functions. In this implementation, two sinusoids are used. Afterwards, the two signals are superimposed, and the resulting signal is the QPSK signal.

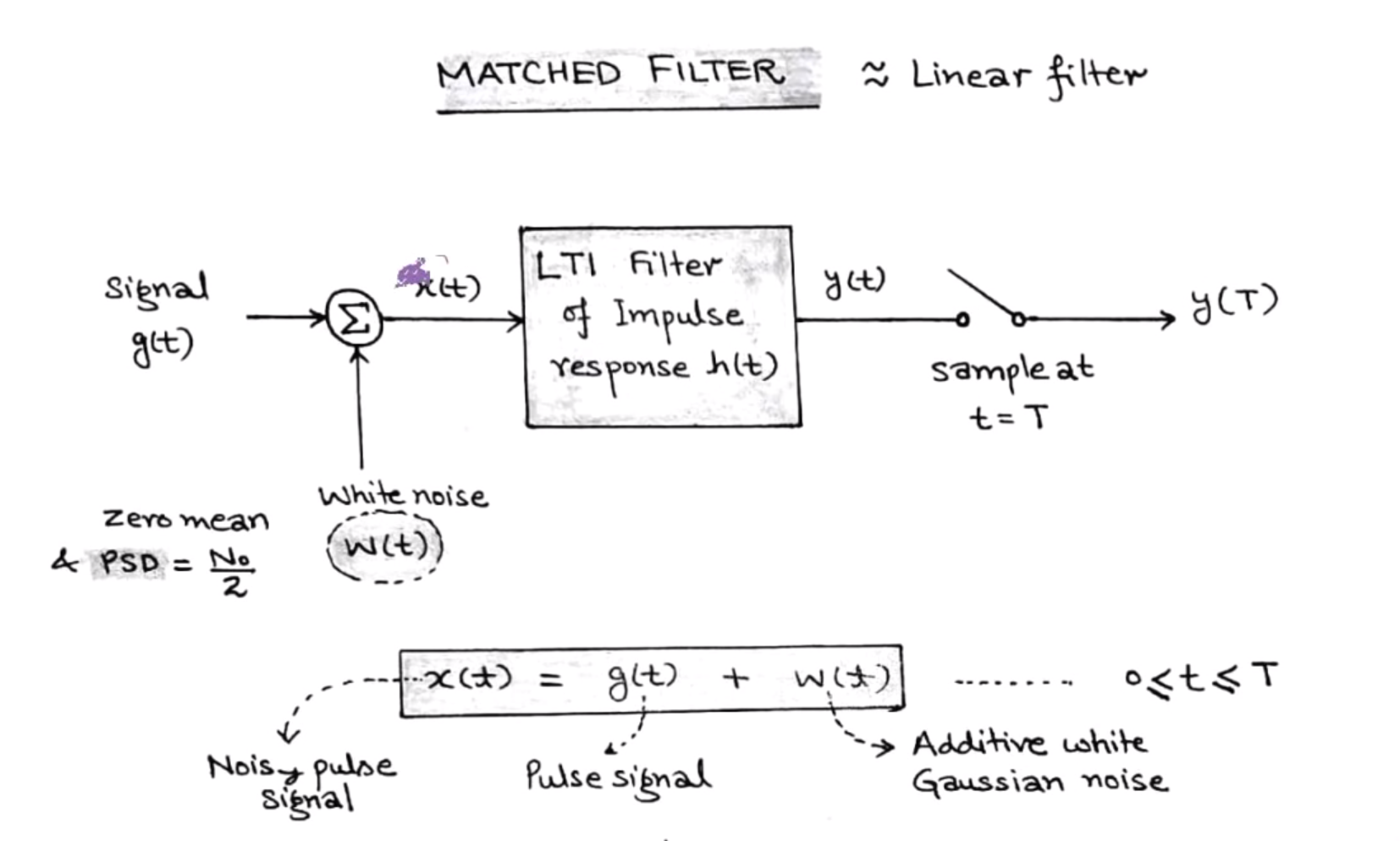
**Channel Bandwidth:** Channel is a medium through which information is transmitted between transmitter and receiver. Channel bandwidth is the frequency range that constitutes the channel. If the message bandwidth is m Hz, then channel bandwidth required to transmit AM is 2m Hz.

Generally, centre frequency is specified and then and say a 'bandwidth of m Hz centered about a frequency fc Hz'

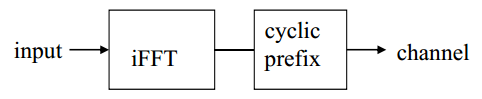
**Matched Filter** (ideal filter)

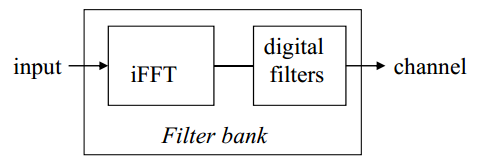
Minimizes effect of noise in received signal

Maximizes SNR of filtered signal

****

**IN ofdma , we apply cyclic prefix guard band**





XIINX Board :

A chip will have various sub components like high performance cpu, low performance cpu, memory, USB,DRAM controller, ethernet ,etc They would need to communicate with each other. This is done through a system called as a bus.. Bus is a centralised part. Siimilar to ROS Master. Or brain. If one component wants to interacts with other, it signals that to the bus and then bus orders it to follow the command.

Master : initiates the transaction

Slave :obeys the transaction

Transaction : transfer of data

Done using 5 imp channels

1)Read transaction data(S-M)

2)Write transaction data(M-S)

3)Read or write transaction address(M-S)

4)read response(S -M) 5)write response(S-M) saying it has written/read

\*Command (M-S)

AXI interconnect

We need a medium through which we can communicate master and slaves.

Called AXI interconnect

Master gives order to AXI interconnect.

Then AXI interconnect gives order to the corresponing slave.

Slave has some address range.

So, if master has to provide order to particular slave it needs to specify its address

1. Capable of making data width conversion. If data width is 64 of master but slave can recieve only 32. AXI interconnect can convert it into two sequences of 32 bits
2. Improves clock freq. Also twp master and slave can work at two diff frequencies but still AXI interconnect helps to transfer the transaction

AXI interconnect can be hierachial

Reason

1. No of ports on a AXI interconnect can be limited. So add an extension

Address range of different address slaves shouldnt overlap

If one slave has address range of 4 kb other has 4 Gb, the address ranges of both slaves cannot be side by side

AXI inteface

Two types :

Memory mapped (5 channels)

Streaming(1 channel)

The data/command actually flows from master to slave. So in streaming inteface we have just 1 channel

Which transfers data

Imp signals

TReady : sent by slave saying it is ready and data can be sent

TValid : if 1 data sent by the master should be processed

Tdata : Signal through which data is sent

Tlast : means end of this frame of data

Naming style

Name of ports

Memory mapped

M\_AXI… S\_AXI…

Stream

M\_AXIS\_… S\_AXIS\_…

If we have a memory map interface we dont have any way t access the address

So suppose we need to access the address to store something we need some interface between them

Done by AXI Datamover

It reads in the command of the stream subsystem and converts it into the memory mapped sub system

If we are recieving data from the stream inteface and converting to memory mapped inteface

AXI datamover will have

An axi slave stream inteface and an axi memory mapped master interface

AXI Memory mapped has 2 types:

Burst and single phase

Single phase

Adress given then only 1 chunk of data sent

Burst

Address given the 1 chunk of data sent

Then another sent at adress+space of prev addresss(like array)

The guy has some tcl script which we can use put in console and write source “name of script file on his site” . in the tcl console

Design AXI based interface using vivado

Microblaze block : our cpu

Has two masters

1. Connected to data bus
2. Connected to instruction bus
3. These two used for booting

On other sidehas axi master interfaces

For data data bus, M\_axi data port ( this port used to talk with other peripherals in ckt),M axi data cache

For instructions, instruction bus or M \_ axi instruction port

Has one interrupt pin

Load micro blaze

Then run block automation

Microblaaze memory port

Stores code used to run

AXI interrupt controller

Gathers interrupts from all parts of the ckt. Sends it to microblaze controller which tells from where interrut has come. Has registers which can mask/cancel do everything with interrupt

AXI UART for debugging

Add AXI timer and UART, to connect them to microblaze connect to interconnect which has one master pin connected to M\_axi data port of controller

To make connections, run connection automation

Concat block : concatenates input signal.

Fior concatenating two interrupt signals

AXI DRAM controller : To store data outtside chip , need dram chips

Connected to cache components of micro blaze through interconnect.

Memory interface generator : used to talk to dram controller located on board

Now i want to connect bram controller to axi microprocessor

This can be done via interconnect

Bram controller is then connected to block memory

Then connect reset and clocks together