Array Like Runtime Reconfigurable MIMO Detector for 802.11n WLAN:A design case study

Pankaj Bhagawat
Rajballav Dash
Gwan Choi
Texas A&M University-CollegeStation

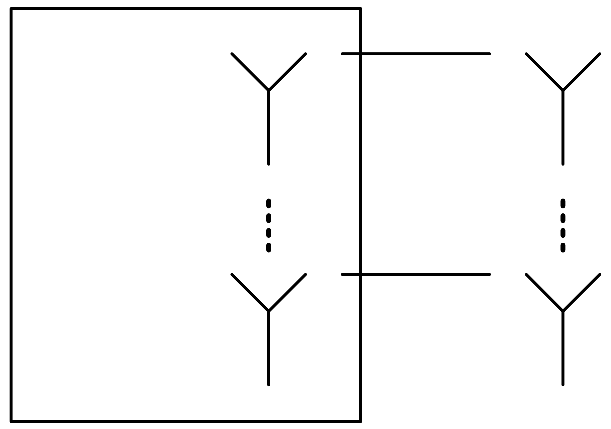
Outline

- Background
- MIMO Detection as a Tree Search
- Related Work
- Fixed Sphere Decoder and Architecture
- Architectural Space Exploration
- Integration Issues in a Communication System

Standards using MIMO and Requirements

- LAN: 802.11n
- WAN: WiMax, LTE etc
- 1Gbps systems like WIGWAM
- All support multiple modulation and coding schemes (MCS)
- Very high throughput requirements at BaseBand
- Ease of integration

Why MIMO?



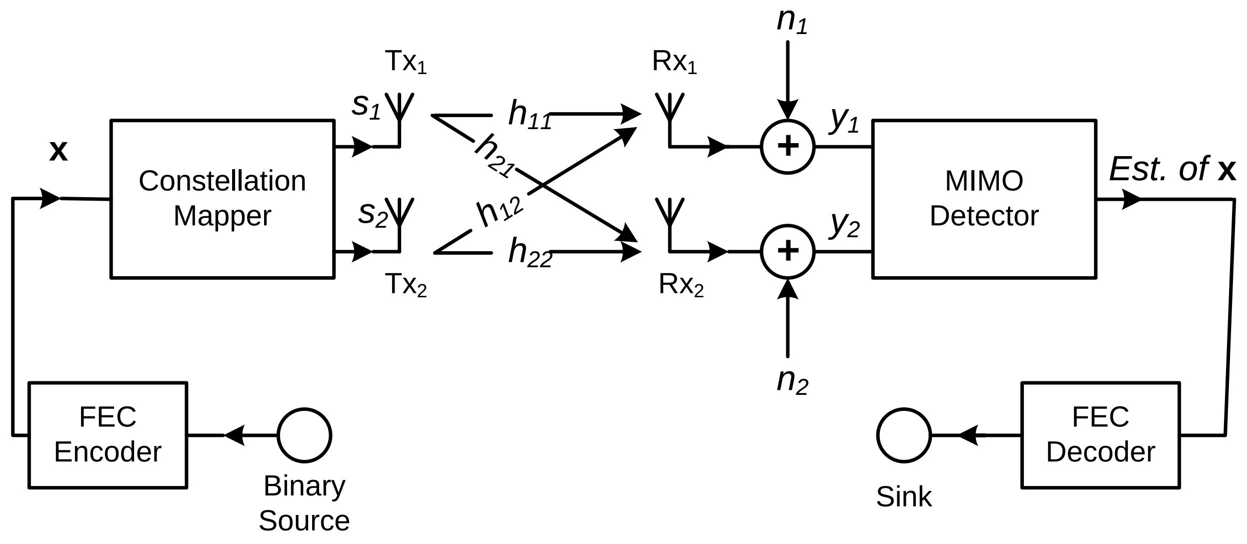
- Spatial Multiplexing is especially attractive
- The transmitter is able to send out multiple data streams on the *same* frequency
- More throughput without extra BW

Ant$_1$

Ant$_n$

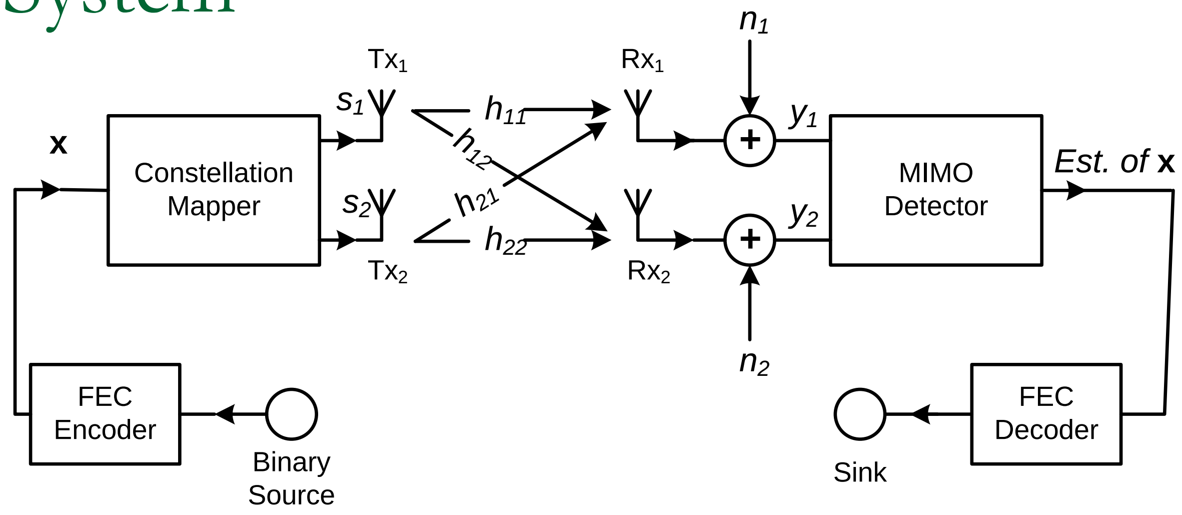
Transmitter

MIMO System



- Binary data is encoded with a rate R code
- Coded bits grouped (in $\log_2 \eta$) and mapped to a $\eta - ary$ QAM
- symbol
- *Independent* QAM symbols transmitted over multiple antennas

MIMO System



- Each Rx antenna sees *weighted superposition* of (or interference from) signals
- from all Tx antennas
  - Rx_1 sees $s_1h_{11}+s_2h_{21}$ and Rx_2 sees $s_1h_{12}+s_2h_{22}$
  - h_{11},h_{12} … are random channel gains (fading)
  - Noise samples n_1 and n_2 further corrupts the interference laden signals

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}$$

- MIMO detection involves removing the interference in presence of noise
- Knowledge of channel gains is assumed (provided by channel estimator)

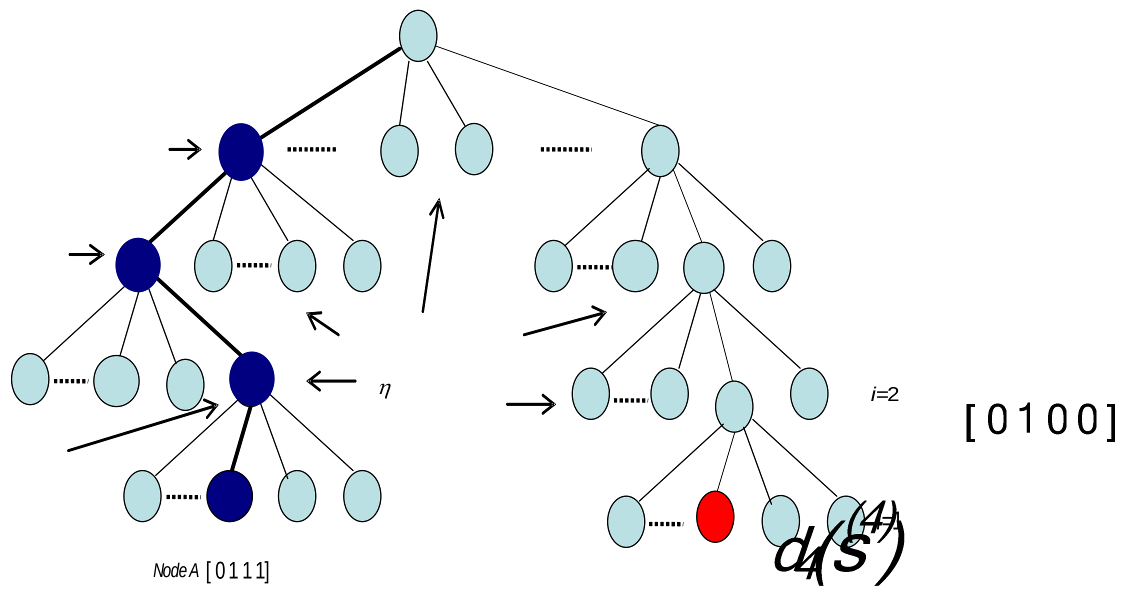
MIMO System

- Can be written in a matrix form

$$\mathrm{y} = \mathrm{Hs} + \mathrm{n}$$

- Best estimate (ML) $\hat{\mathbf{s}}$ of **s** is such that it minimizes $\left\| \mathbf{y} - \mathbf{Hs} \right\|^2$
- **H**=**QR** **,** **R** is upper triangular matrix. Pick **s** such that $\left\| \hat{\mathbf{y}} - \mathbf{Rs} \right\|^2$ is minimized

Tree Structure for 4x4-16 QAM MIMO System



$$d_i(s^{(i)}) = d_{i+1}(s^{(i+1)}) + |e_i(s^{(i)})|^2$$

$$|e_i(s^{(i)})|^2 = |c_{i+1}(s^{(i+1)}) - R_{ii}.s_i|^2$$

$$c_{i+1}(s^{(i+1)}) = \hat{y}_i - \sum_{j=i+1}^{M_T} R_{ij}.s_j$$

- Due to the upper triangular nature of **R**, best estimate of **s** can be treated as a tree search

[1 1 0 1]

$d_3(s^{(3)})$

- Incremental metric $|e_i|$, is always positive
- c_{i+1} ,and hence $|e_i|$, depends only path history and the present QAM symbol s_i

[0 1 0 1]

Each node has

=16 children

- Path corresponding to the leaf node with least *d* corresponds to the best *hard* estimate of **s** (ML)

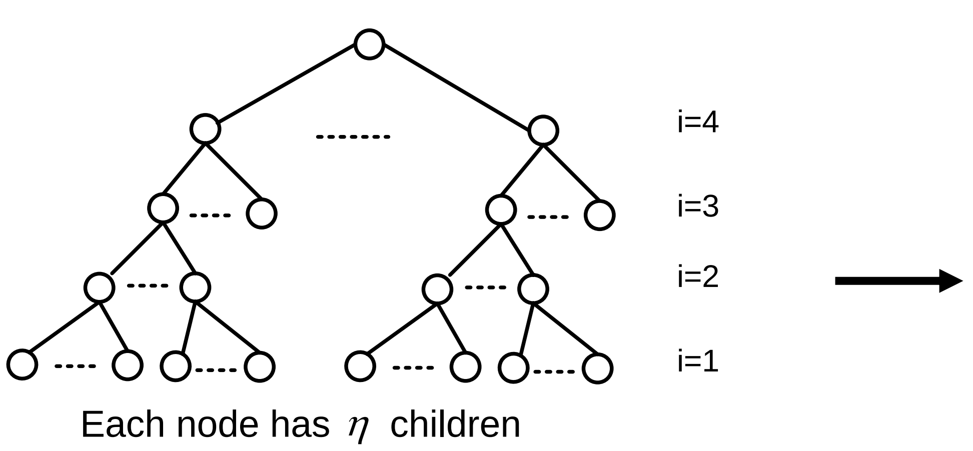
$d_2(s^{(2)})$

$d_1(s^{(1)})$

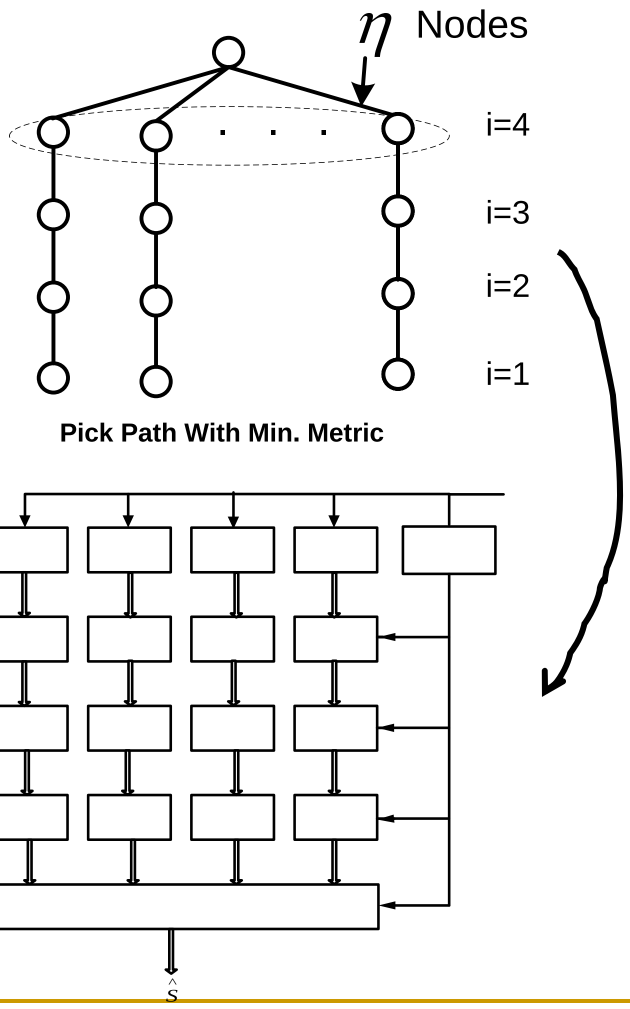
Related Work

- K-best algorithm takes a BFS approach and retains K “best” paths at each level of the tree
  - Fixed throughput can be achieved
  - Sorting is very expensive
  - Large memory to store intermediate results depending on the value of “K”
  - The value of “K” is modulation scheme dependent=> difficult to achieve high resource utilization in a reconfigurable environment
- DFS based approach : Searches the tree in a depth first manner
  - Highly random throughput, throughput not high enough
  - Hard to parallelize and pipeline due to a feedback loop
- Linear Detectors are low complexity but has poor BER/FER

FSD Algorithm



Each node has η children



$$d_4\left(\mathbf{s}^{(4)}\right) = \left\| y_4 - R_{44}.s_4 \right\|^2$$

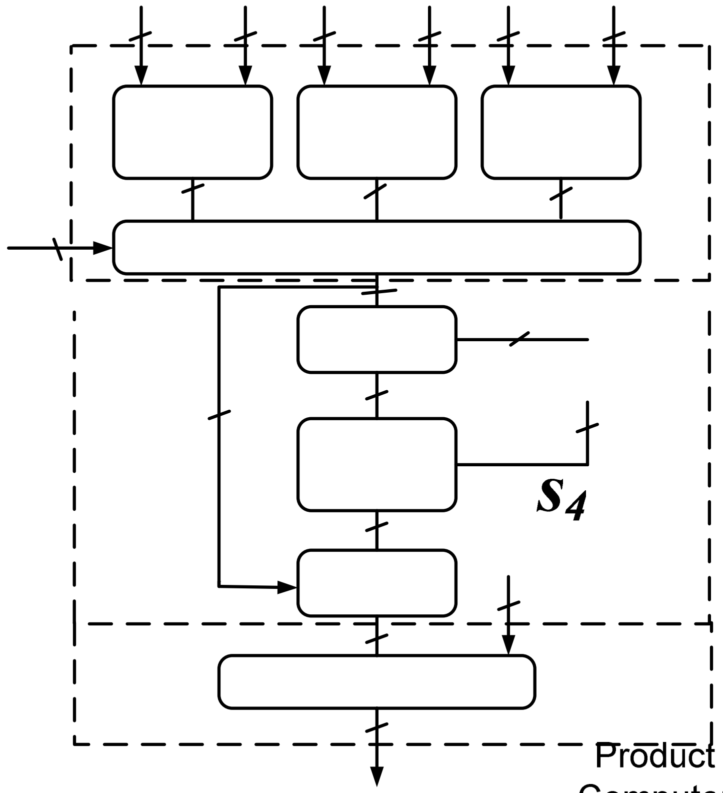
$$d_3\left(\mathbf{s}^{(3)}\right) = d_4\left(\mathbf{s}^{(4)}\right) + \left\| y_3 - R_{34}.s_4 - R_{33}.s_3 \right\|^2$$

$$d_2\left(\mathbf{s}^{(2)}\right) = d_3\left(\mathbf{s}^{(3)}\right) + \left\| y_2 - R_{24}.s_4 - R_{23}.s_3 - R_{22}.s_2 \right\|^2$$

$$d_1\left(\mathbf{s}^{(1)}\right) = d_2\left(\mathbf{s}^{(2)}\right) + \left\| y_1 - R_{12}.s_2 - R_{13}.s_3 - R_{14}.s_4 - R_{11}.s_1 \right\|^2$$

Metric Computation Unit (MCU)

$$d_1\left(\mathbf{s}^{(1)}\right) = d_2\left(\mathbf{s}^{(2)}\right) + \left\| y_1 - R_{12}.s_2 - R_{13}.s_3 - R_{14}.s_4 - R_{11}.s_1 \right\|^2$$



$$c_{i+1}(\mathbf{s}^{(i+1)}) = \hat{y}_i - \sum_{j=i+1}^{M_T} R_{ij}.s_j$$

$$\left|e_i(\mathbf{s}^{(i)})\right|^2 = \left|c_{i+1}(\mathbf{s}^{(i+1)}) - R_{ii}.s_i\right|^2$$

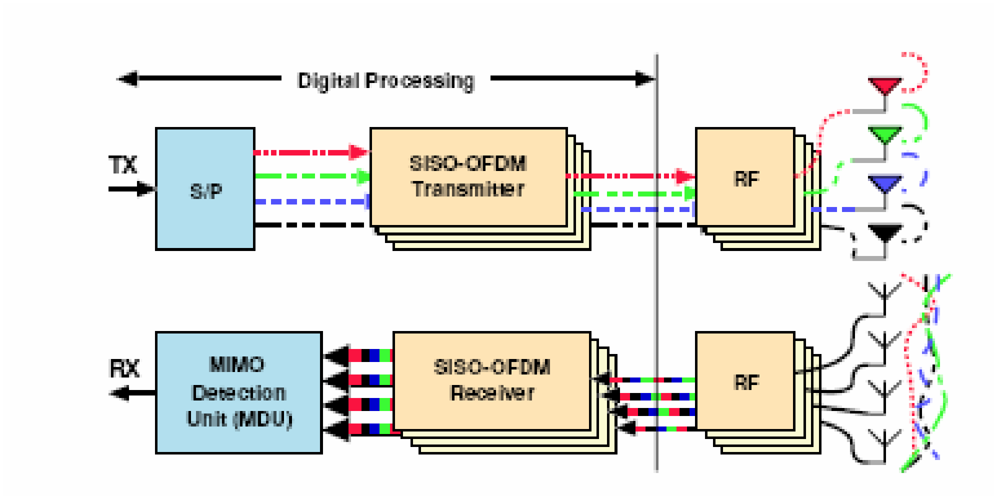
$$d_i(\mathbf{s}^{(i)}) = d_{i+1}(\mathbf{s}^{(i+1)}) + \left|e_i(\mathbf{s}^{(i)})\right|^2$$

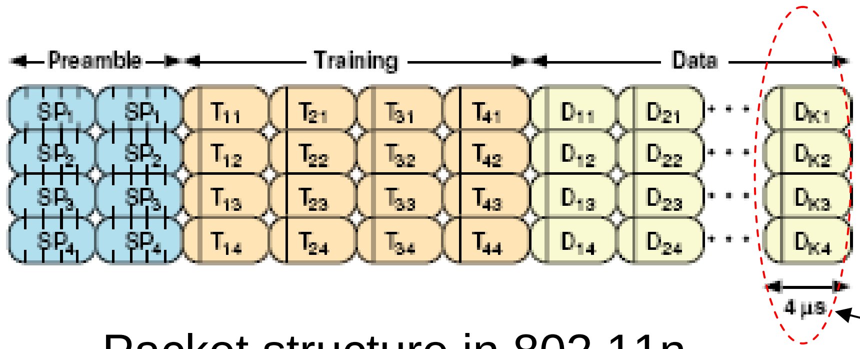
- Product Computer: Computes the products $R_{ij}s_j$
- Slicer finds the best child, MF[1:0] configures the slicer to operate for different modulation schemes

Key Features

- Systolic type array of processing elements
- On the fly reconfiguration possible
- Highly pipeline-able
- Data and control flow is forward flowing
- Fixed throughput for a given modulation scheme
- Very high resource utilization
  - MCUs *matched* to level of the tree
  - Pipeline is not broken

MIMO-OFDM System(Interface)**



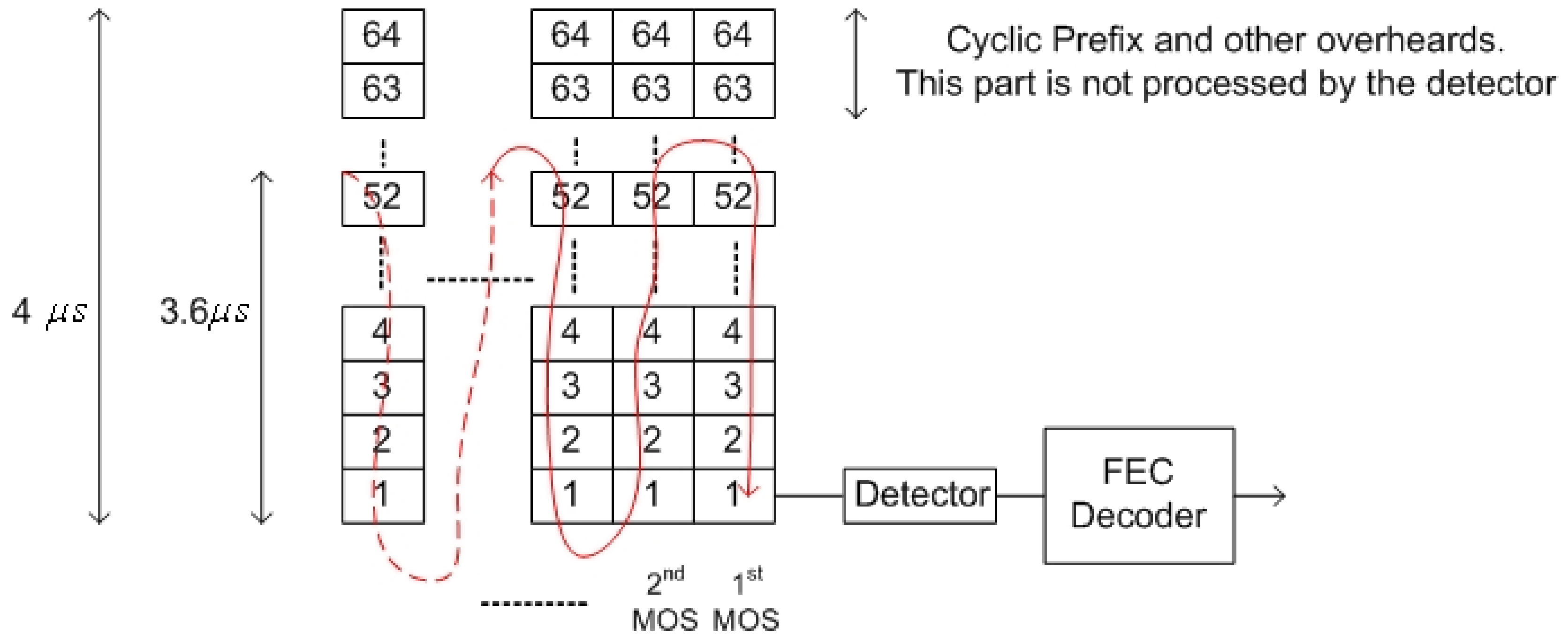


Preamble
Training
Data
SP1 SP2 SP3 SP4
T11 T12 T13 T14 T21 T22 T23 T24 T31 T32 T33 T34 T41 T42 T43 T44
D11 D12 D13 D14 D21 D22 D23 D24
DK1 DK2 DK3 DK4
4 µs
MIMO-OFDM Symbol

Packet structure in 802.11n type systems

** *Perels, D. et. al. "ASIC Implementation* of a *MIMO-OFDM*. Transceiver for 192 Mbps WLANs." ESSCIRC 2005

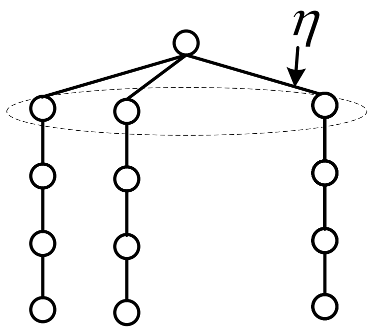
MIMO-OFDM System

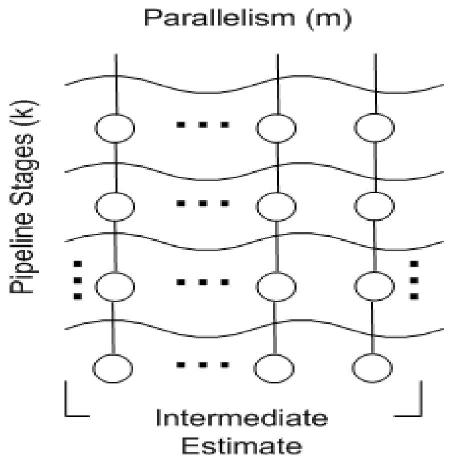


64
63
52
4
3
2
1
4 μs
3.6μs
Cyclic Prefix and other overheards.
This part is not processed by the detector
2nd MOS
1st MOS
Detector
FEC Decoder

- MOS=MIMO-OFDM Symbol
- A detector core has to process 52 tones in 3.6microsecs.

Detector Array





Nodes

- Processing time for 52 tones, T_p, is given by $T_p = 52\lceil \eta / m \rceil C_d / (k+1)$
- η depends on the modulation format used, η=4(QPSK), 16(16-QAM), 64(64-QAM).
- C_d is the combinational delay of the un-pipelined array.

. . .

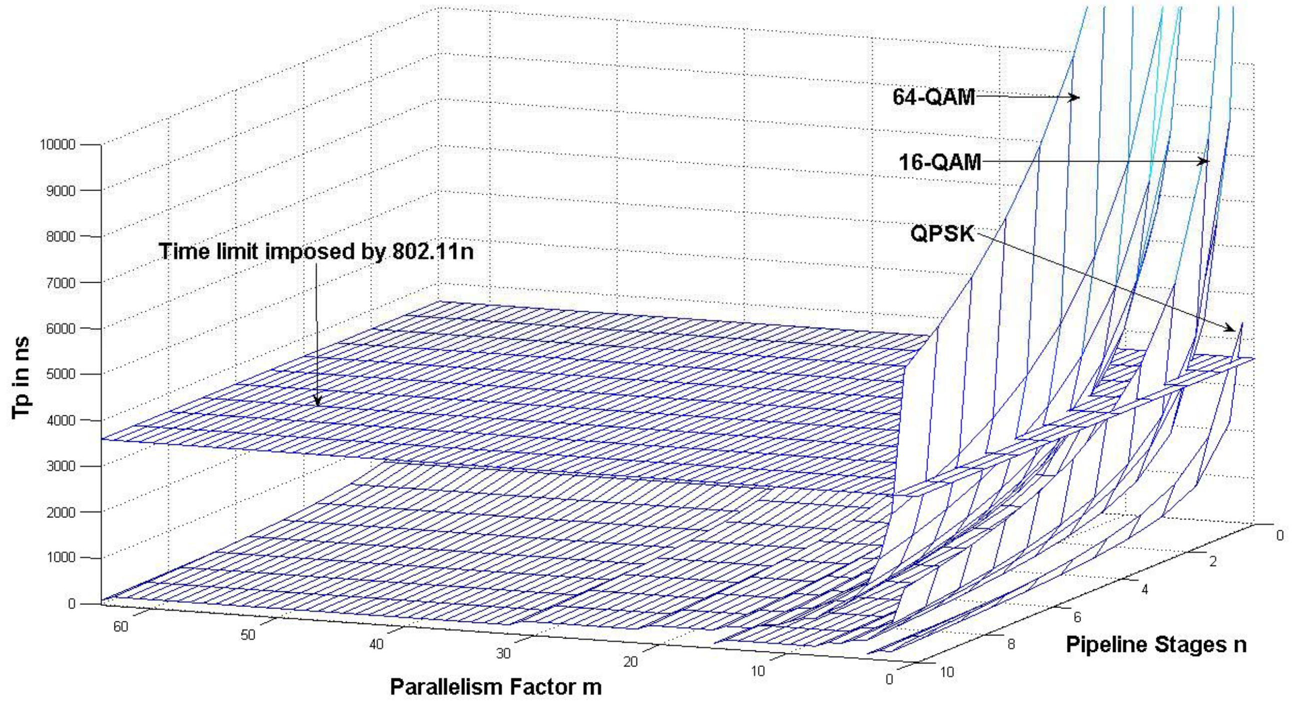
i=4

i=3

i=2

i=1

Architectural Space



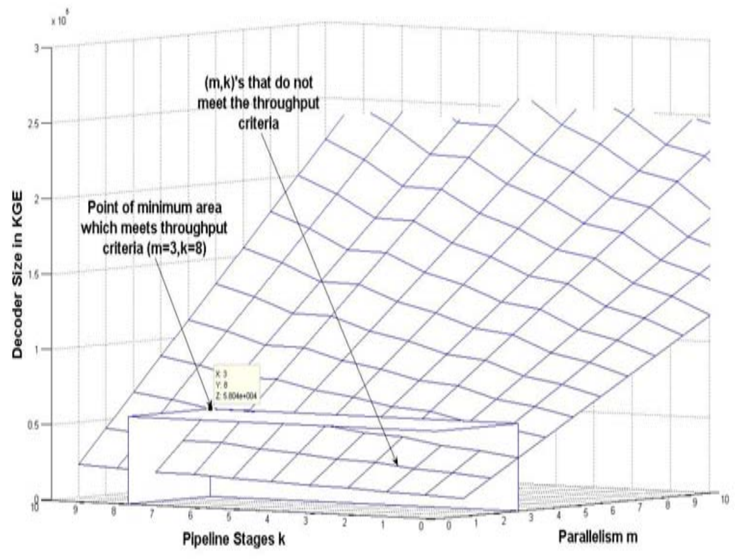
- In actual simulations T_p = 3000ns (to account for 15-20% pessimism factor).

Power, Delay, and Area Estimation

- Power consumption mainly due to logic and clock network
- Clock network is modeled as a symmetrical mesh.
  - Global clock network power was estimated using HSPICE
  - Local power is estimated using capacitive load due to number of flops driven by a local clock buffer
- Synopsys DC was used to estimate logic power,area, and timing of the detector
- DC retiming utility was used to introduce and retime the pipes

Architectural Exploration for Low Area

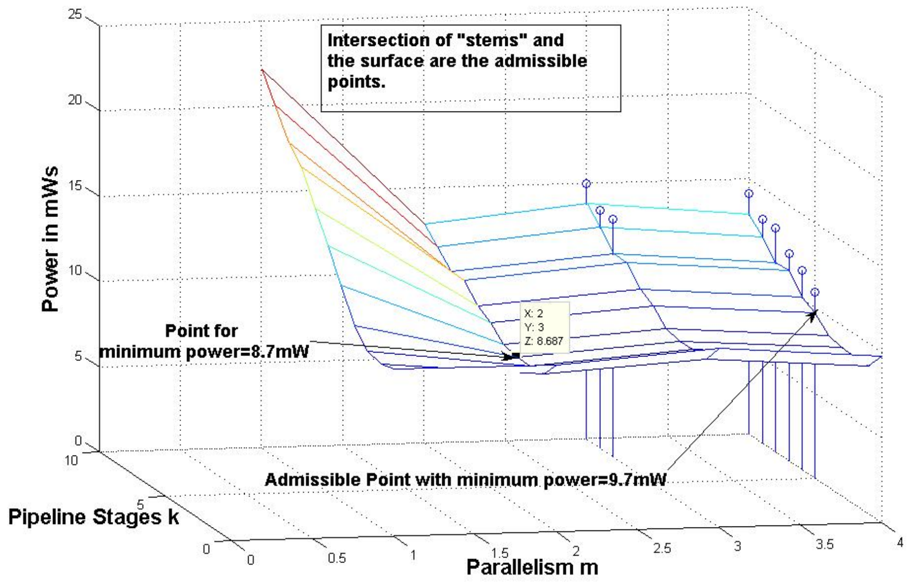
- Find (m,k) such that it meets the throughput requirements for all modes and has minimum area
- 64-QAM is most intensive=>find (m,k) to meet throughput requirement for it
- M=3, and k=8 is able to meet the requirement with lowest area



Architectural Exploration for Low Power

- Power consumption profiles differ with the modulation modes
- Power consumption has to be optimized over all the modes
- Find (m,k) such that the aggregate power is minimized and detector still meet the throughput requirements

Architectural Exploration for Low Power



- Aggregate power $Pow_{agg} = Prob(\text{QPSK})*Pow(\text{QPSK}) + Prob(\text{16QAM})*Pow(\text{16QAM}) + Prob(\text{64QAM})*Pow(\text{64QAM})$
- All the probabilities can be assumed =1/3, since there is no a-priori knowledge
- Pow_{agg} as a function of (m,k) is shown in the figure
    - Not all points meet the throughput req.
    - Admissible points are shown as stems.
    - m=4 and k=5 => lowest power while meeting throughput req.

Results

OUR ASIC IMPLEMENTATION DETAILS

Design Parameters	Area Optimized	Power Optimized
Target Tech. Library	Nangate 45nm PDK	Nangate 45nm PDK
Pipeline Stages (k)	8	5
Parallelism (m)	3	4
Gate Equivalent	58.2k	67.7k
Power Consumption	11.91mW	9.7mW
Frequency: QPSK	38.8MHz	18MHz
Frequency: 16-QAM	116.3MHz	71.8MHz
Frequency: 64-QAM	426.6MHz	287.3MHz
Throughput Requirement: QPSK	115.6Mbps	115.6Mbps
Throughput Achieved: QPSK	155Mbps	144Mbps
Throughput Requirement: 16-QAM	231.1Mbps	231.1Mbps
Throughput Achieved: 16-QAM	310.13Mbps	287.2Mbps
Throughput Requirement: 64-QAM	346.7Mbps	346.7Mbps
Throughput Achieved: 64-QAM	465.38Mbps	430.95Mbps

Conclusion & Future Work

- First design pushes for least area
- Second one tries to achieve least power
- Detector is configurable depending the modulation scheme used
- Future work will focus on architectures to support different *sized* MIMO systems(2x2,3x3,4x4 etc), this finds application in a multi-protocol device or SDR

Thank You!!