

Binary Source

November 11, 2016

This block generates a sequence of binary values (1 or 0) and it can work in four different modes:

1. Random
2. PseudoRandom
3. DeterministicCyclic
4. DeterministicAppendZeros

This blocks doesn't accept any input signal. It produces any number of output signals.

Input Parameters

- mode{PseudoRandom}
(Random, PseudoRandom, DeterministicCyclic, DeterministicAppendZeros)
- probabilityOfZero{0.5}
(real $\in [0,1]$)
- patternLength{7}
(integer $\in [1,32]$)
- bitStream{"0100011101010101"}
(string of 0's and 1's)
- numberOfBits{-1}
(long int)
- bitPeriod{1.0/100e9}
(double)

Methods

```
BinarySource(vector<Signal *> &InputSig, vector<Signal *> &OutputSig) :Block(InputSig, OutputSig){};
```

```
void initialize(void);
```

```
bool runBlock(void);
```

```
void setMode(BinarySourceMode m) mode = m; BinarySourceMode const getMode(void) { return mode; };
```

```
void setProbabilityOfZero(double pZero) { probabilityOfZero = pZero; };
```

```
double const getProbabilityOfZero(void) { return probabilityOfZero; };
```

```
void setBitStream(string bStream) { bitStream = bStream; };
```

```
string const getBitStream(void) { return bitStream; };
```

```
void setNumberOfBits(long int nOfBits) { numberOfBits = nOfBits; };
```

```
long int const getNumberOfBits(void) { return numberOfBits; };
```

```
void setPatternLength(int pLength) { patternLength = pLength; };
```

```
int const getPatternLength(void) { return patternLength; }
```

```
void setBitPeriod(double bPeriod);
```

```
double const getBitPeriod(void) { return bitPeriod; }
```

Functional description

The *mode* parameter allows the user to select between one of the four operation modes of the binary source.

Random Mode Generates a 0 with probability *probabilityOfZero* and a 1 with probability $1 - \text{probabilityOfZero}$.

Pseudorandom Mode Generates a pseudorandom sequence with period $2^{\text{patternLength}} - 1$.

DeterministicCyclic Mode Generates the sequence of 0's and 1's specified by *bitStream* and then repeats it.

DeterministicAppendZeros Mode Generates the sequence of 0's and 1's specified by *bitStream* and then it fills the rest of the buffer space with zeros.

Input Signals

Number: 0

Type: Binary (DiscreteTimeDiscreteAmplitude)

Output Signals

Number: 1 or more

Type: Binary (DiscreteTimeDiscreteAmplitude)

Examples

Random Mode

PseudoRandom Mode As an example consider a pseudorandom sequence with *patternLength*=3 which contains a total of 7 ($2^3 - 1$) bits. In this sequence it is possible to find every combination of 0's and 1's that compose a 3 bit long subsequence with the exception of 000. For this example the possible subsequences are 010, 110, 101, 100, 111, 001 and 100 (they appear in figure 1 numbered in this order). Some of these require wrap.

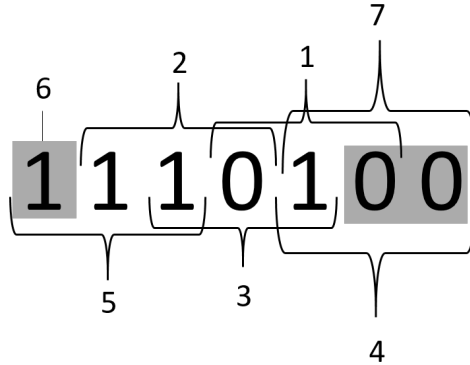


Figure 1: Example of a pseudorandom sequence with a pattern length equal to 3.

DeterministicCyclic Mode As an example take the *bit stream* '0100011101010101'. The generated binary signal is displayed in.

DeterministicAppendZeros Mode Take as an example the *bit stream* '0100011101010101'. The generated binary signal is displayed in 2.

Suggestions for future improvement

Implement an input signal that can work as trigger.

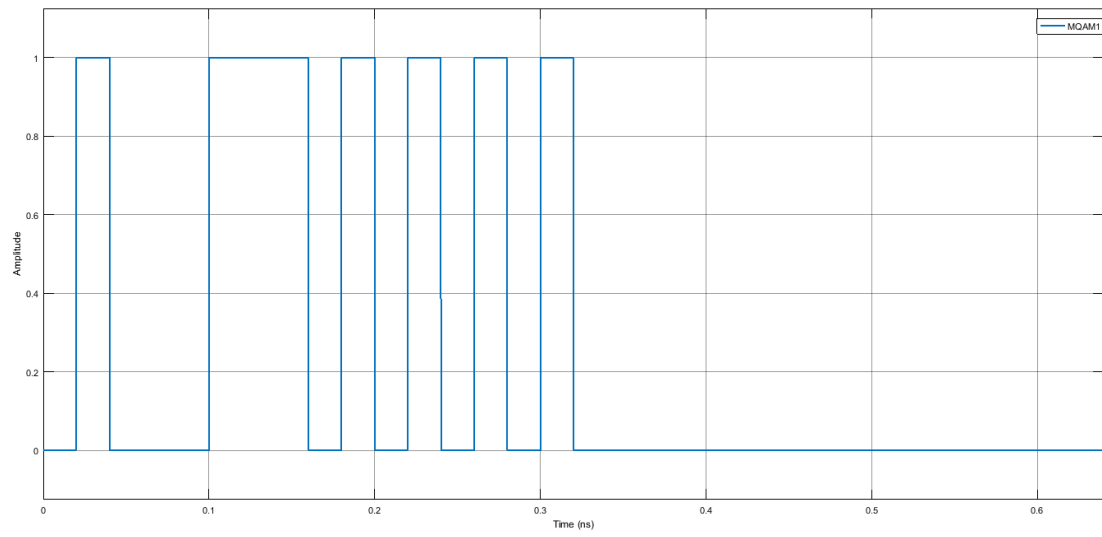


Figure 2: Binary signal generated by the block operating in the *PseudoRandom* mode