ELEC1200: A System View of Communications: from Signals to Packets Lecture 2

- Recap of last time:
  - Bit sequences
  - Representing bit sequences as waveforms
- Discrete time waveforms
- Representing discrete time bit waveforms
  - Unit step function
  - Sums of unit step functions
  - Equivalent representations of waveforms

Bit Sequences

- Information we want to send can be encoded as long bit sequences created by concatenating binary code words.

$b_7 = 0$

USASCII code chart

Bits $b_6 b_5 b_4$ →					000	001	010	011	100	101	110	111
b_3	b_2	b_1	b_0	Row ↓ / Column →	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	|
1	1	0	1	13	CR	GS	−	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

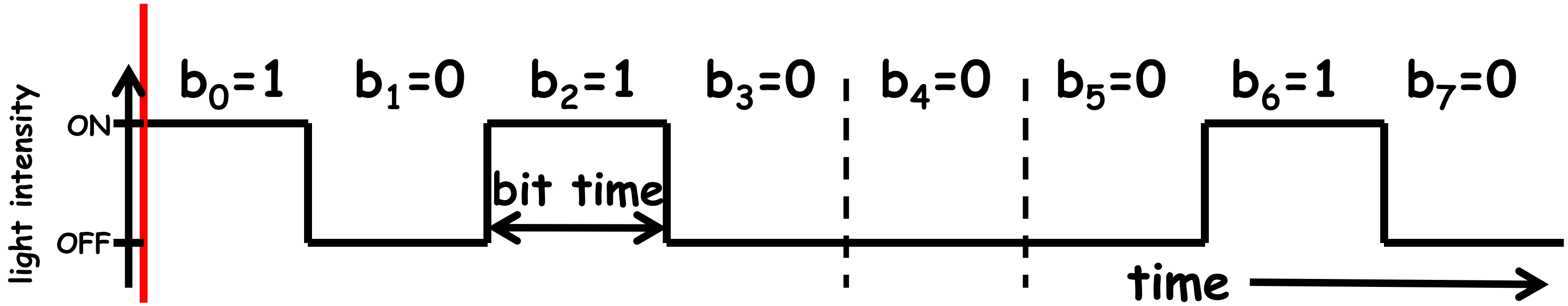
$b_0\ b_1\ b_2\ b_3\ b_4\ b_5\ b_6\ b_7$

1 0 1 0 0 0 1 0 | 1 1 0 0 0 0 1 0 | 1 0 1 0 0 0 1 0

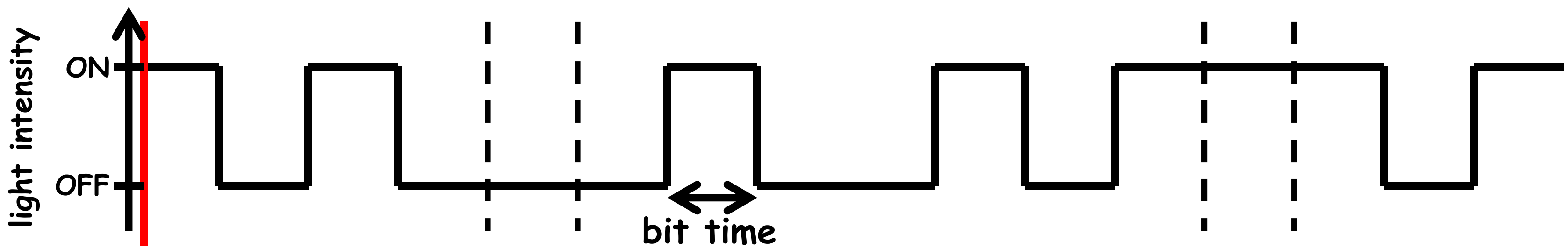
E | C | E

Representing Bit Sequences as Waveforms

- A bit sequence can be encoded by changing the value of the physical variable over time.



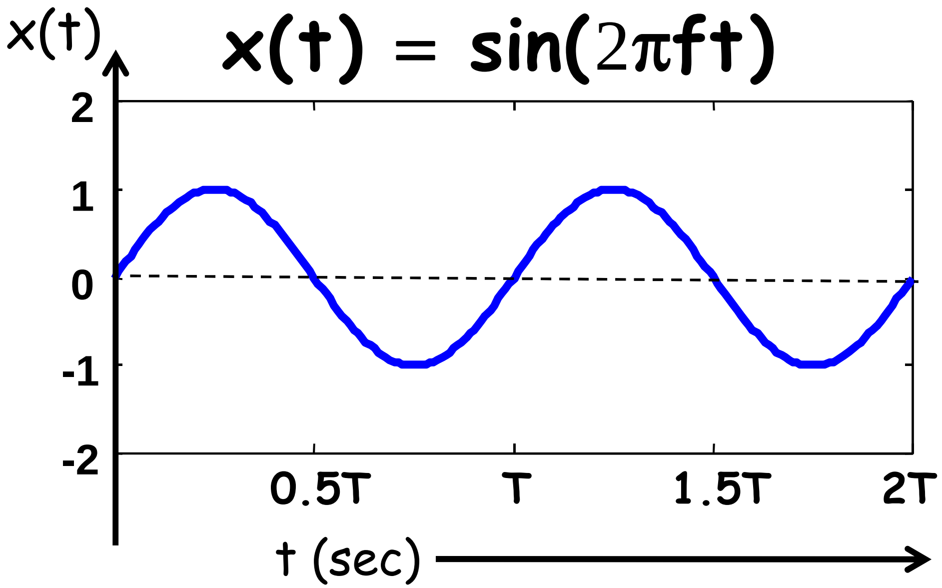
- Each bit is encoded by holding the state constant over a length of time, known as the bit time.
- The shorter the bit time, the faster we can transmit information (bits)



Representing Signals or Waveforms

- Mathematically, a signal can be represented as a function of one or more variables, e.g. time
- Example:

$$x(t) = \sin(2\pi f t)$$

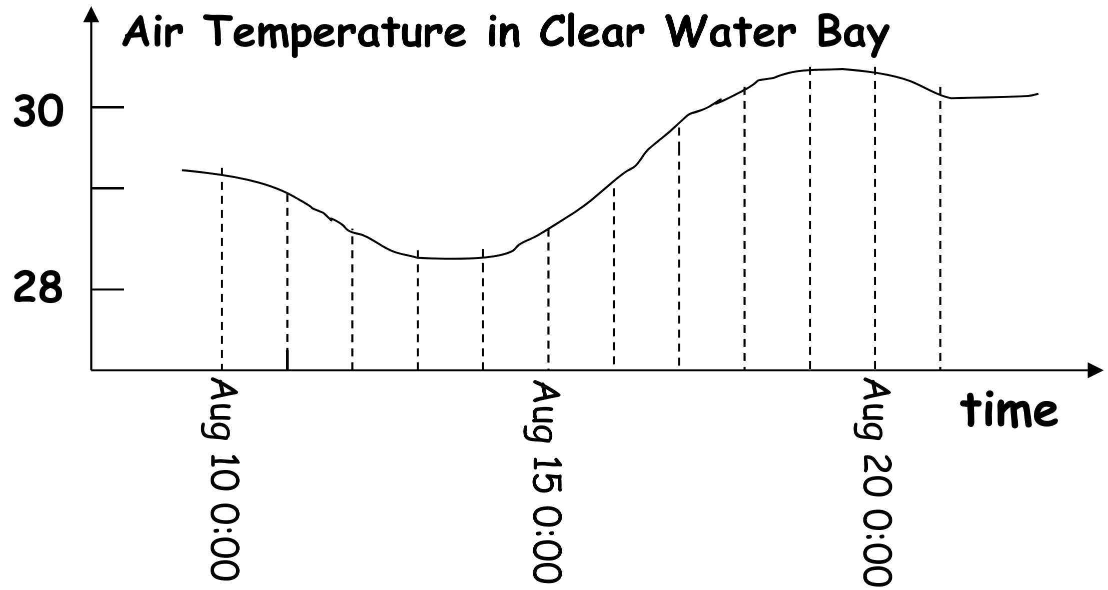


t = time (in seconds)
T = period (in seconds)
f = frequency in hertz (cycles per second)

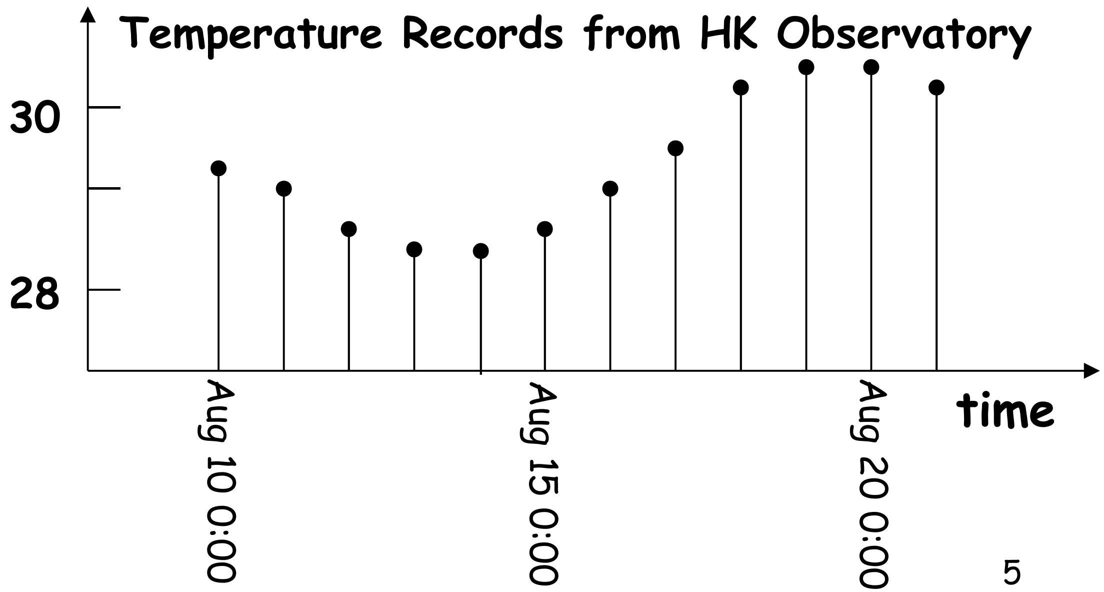
$$T = \frac{1}{f}$$

Continuous and Discrete Time Signals

- A Continuous Time (CT) signal has a known value for <u>all</u> points in a time interval



- A Discrete Time (DT) signal has a known value only at a discrete (discontinuous) set of time points.

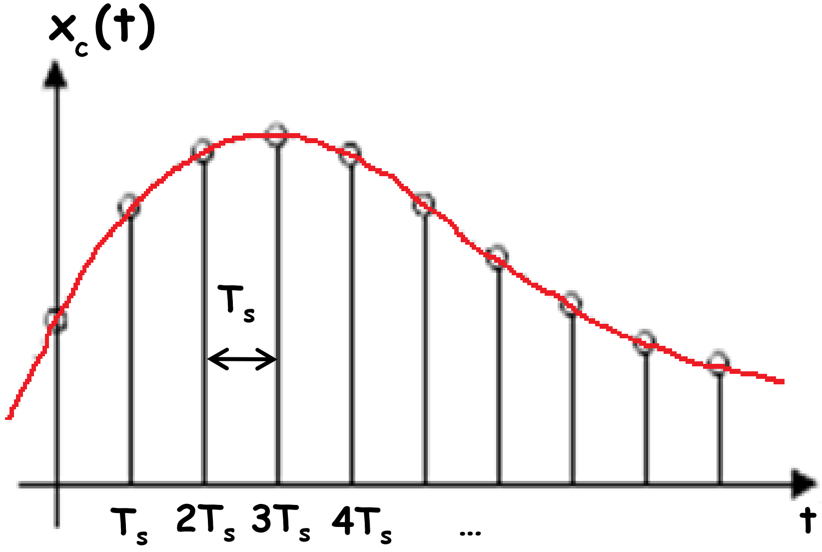


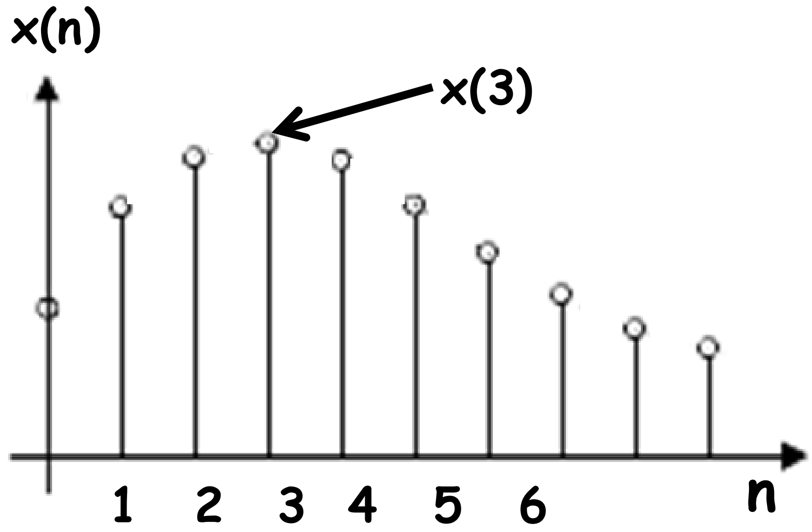
From continuous to discrete time: sampling

- We can obtain discrete time waveform by <u>sampling</u> (recording) a continuous time waveform $x_c(t)$ at regular intervals in time.
  - T_s = <u>sample period</u>
- We typically index (identify, label) each sample by an integer sample number, n.
- We denote the sampled waveform by x(n)
- The nth sample corresponds to the waveform at time

$$t = nT_s$$

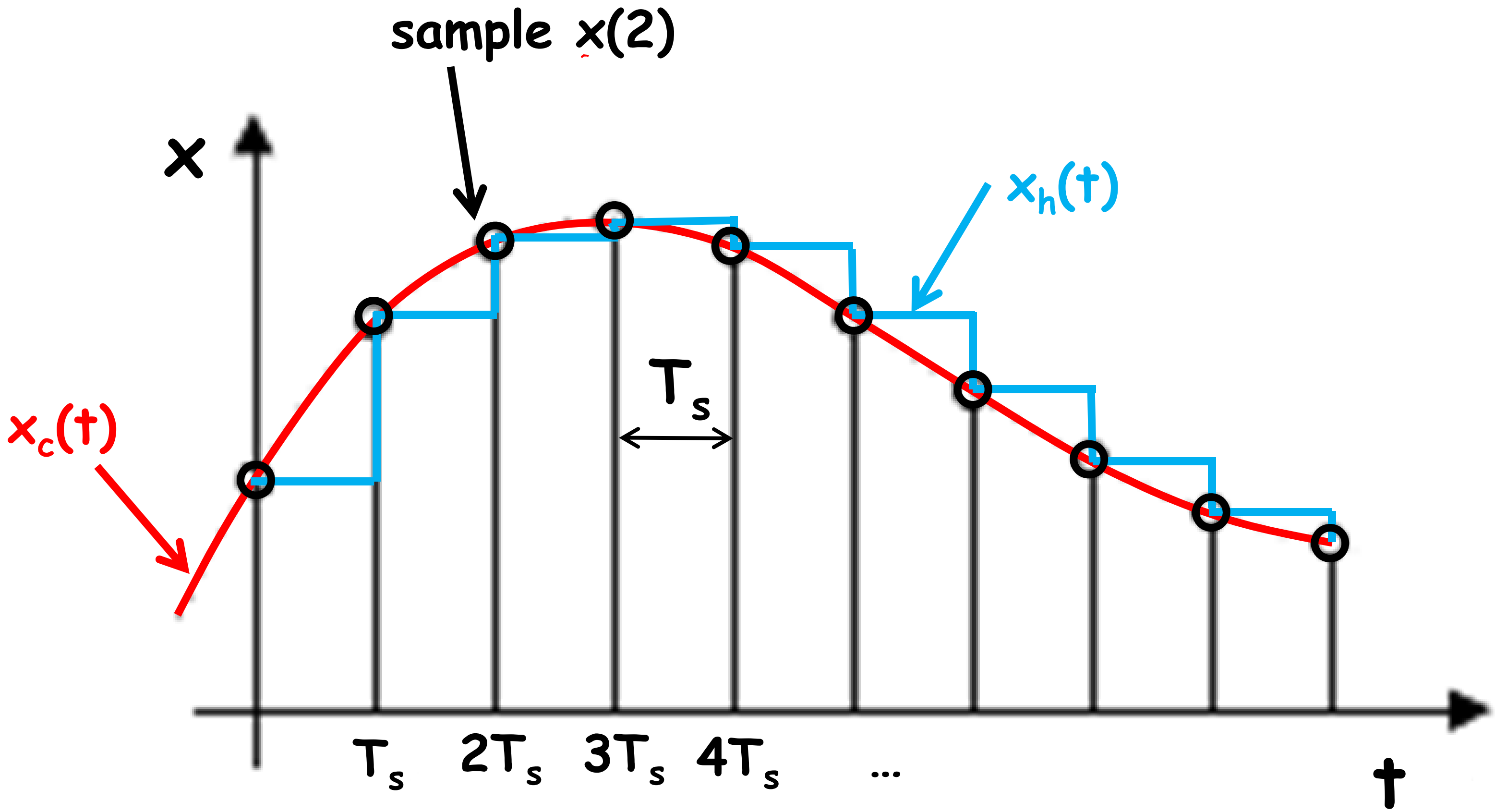
i.e. $x(n) = x_s(nT_s)$





From discrete to continuous time

- Given samples x(n), we can obtain a continuous time waveform $x_h(t)$ by "holding" the waveform at x(n) between times nT_s, and $(n+1)T_s$



Sampling period vs. frequency

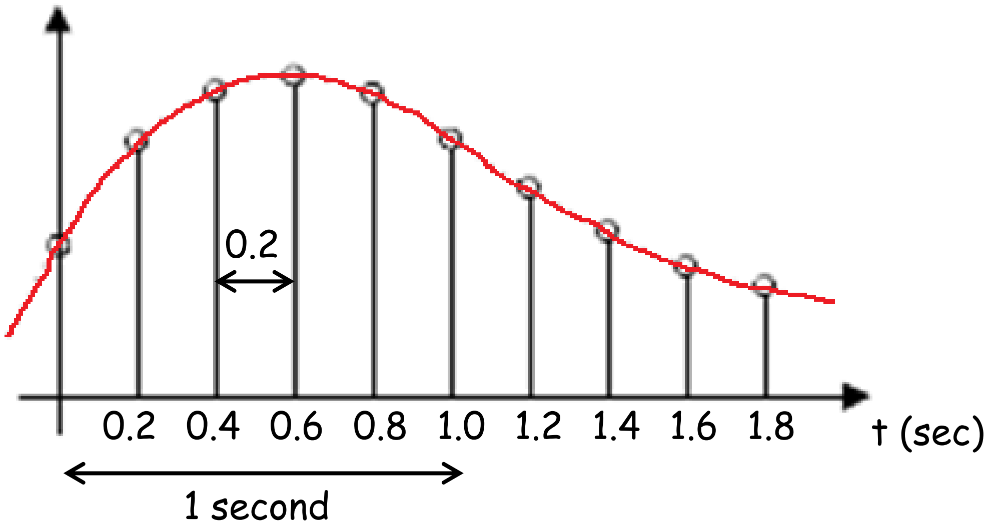
- T_s = <u>sample period</u> (time interval between samples)
  - Typical unit: seconds (s, sec)
- F_s = <u>sampling frequency or rate</u> (number of samples in a fixed period of time)
  - Typical unit: Hertz (Hz, samples per second)

- Relationship: $F_s = \frac{1}{T_s}$

- Example:

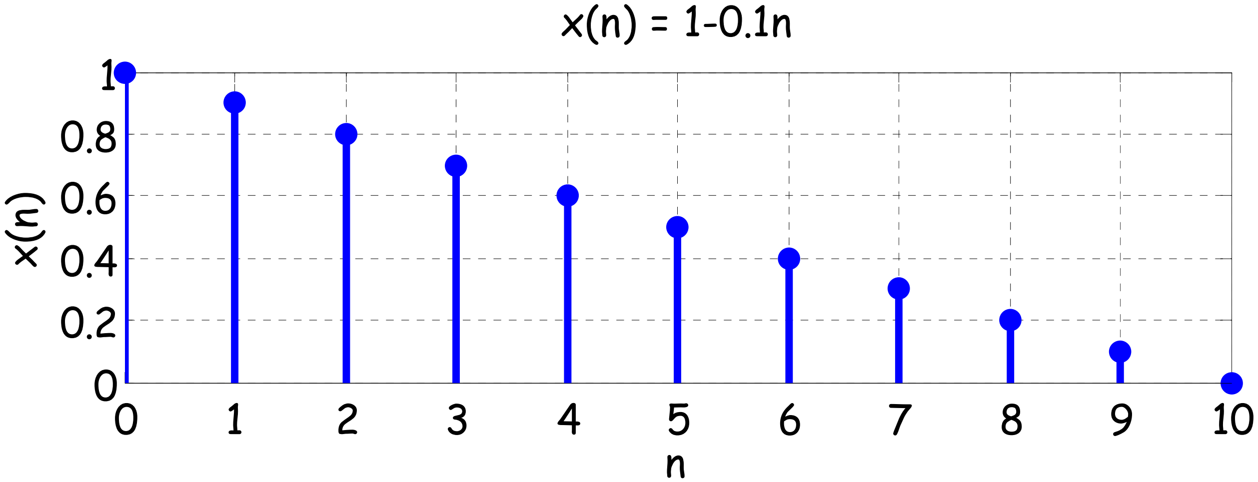
$$T_s = 0.2 \text{ sec}$$

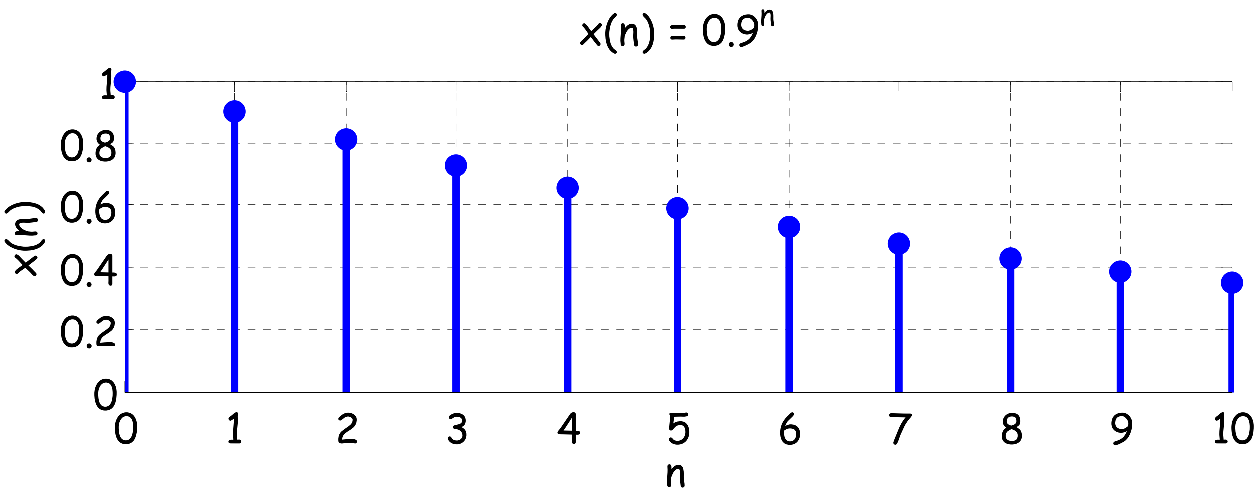
$$F_s = \frac{1 \text{ sample}}{0.2 \text{ sec}} = 5 \frac{\text{sample}}{\text{sec}} = 5\text{Hz}$$



Mathematical Representations

- Mathematically, we define sampled signals as functions of n. We can represent these functions using formulas or graphs. Each is useful, but in different ways.

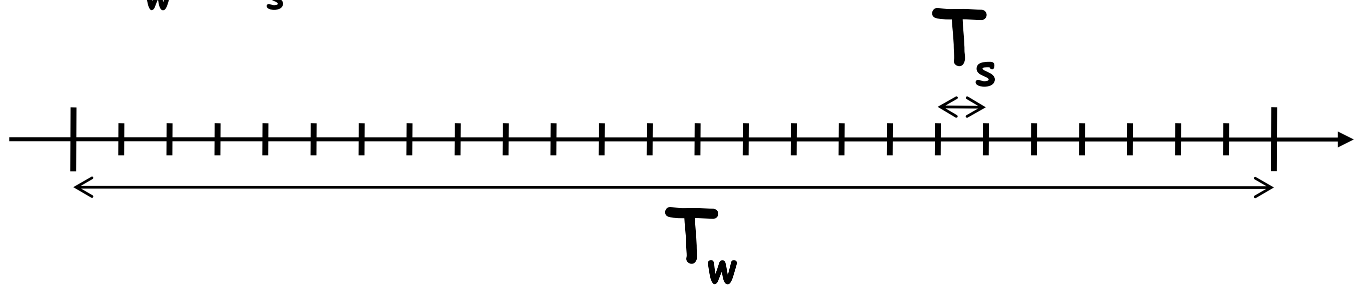




Number of samples

- If we sample a signal at intervals of T_s over a finite time window T_w, we obtain N samples where

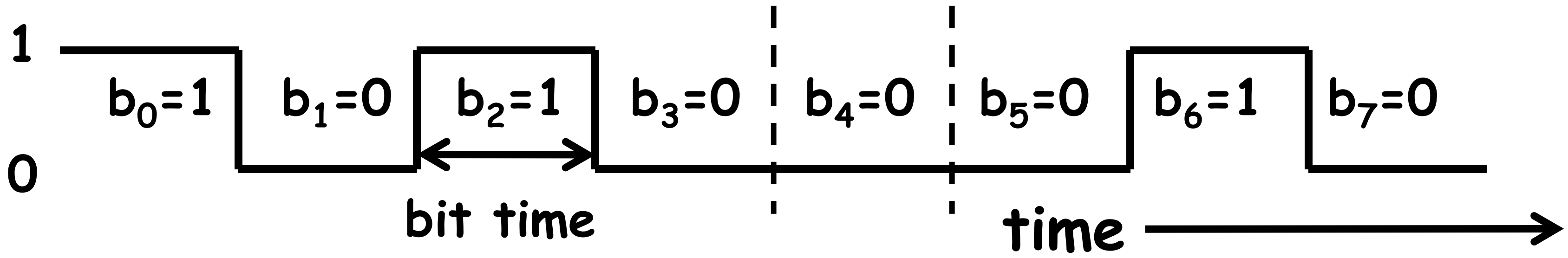
$$N = \frac{T_w}{T_s} = T_w \cdot F_s$$



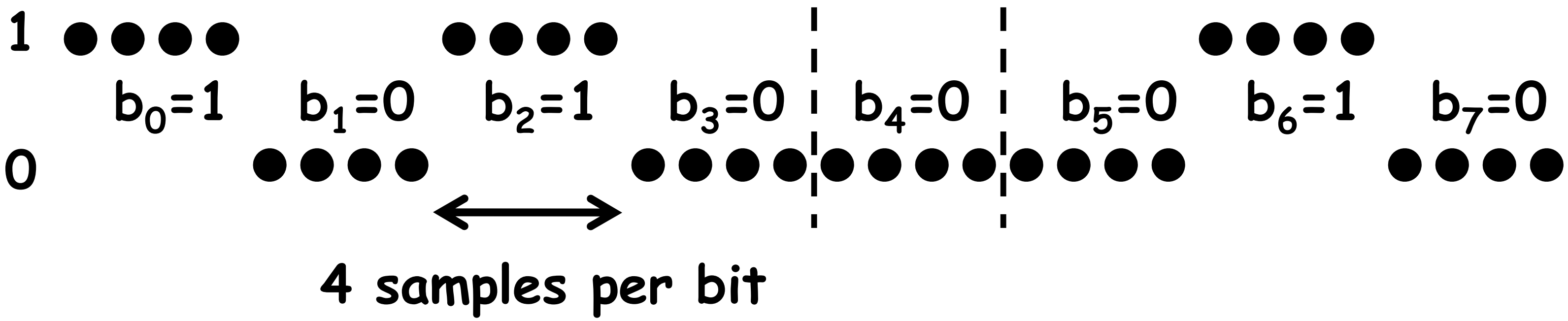
- This is an example of an engineering <u>tradeoff</u>: a higher sample frequency is
  - Good: Less information lost since less time between samples
  - Bad: More storage needed since more samples for a given length of time.

From Bit Sequences to Bit Waveforms

- Recall that bits can be encoded by holding a physical variable constant over a "bit time."



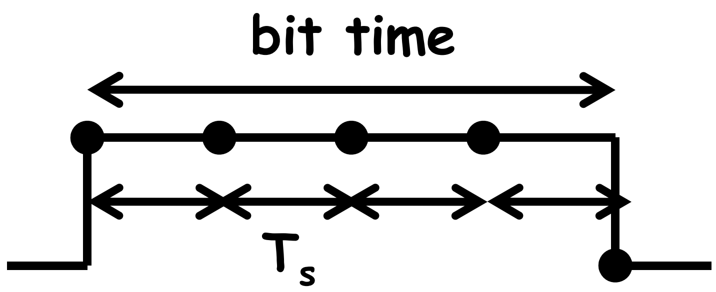
- In discrete time, we encode bits as waveforms by holding sample values constant over a number of "samples per bit" (SPB)



Bit Rate, Sampling Frequency, SPB

- The bit rate measures the number of bits we can send in a given unit of time. We generally want this to be as large as possible.
- We get faster bit rates by
  - Increasing sample rate (F_s), or
  - Decreasing samples per bit (SPB)
- <u>In this course, we will always use the same sample rate</u>

$$
\begin{aligned}
F_s &= 1\text{MHz} = 1 \text{ MegaHertz} \\
&= 1,000,000 \text{ samples/second} \\
&= 10^6 \text{ samples/second} \\
T_s &= 1\mu\text{sec} = 1 \text{ microsecond} \\
&= 0.000001 \text{ second} \\
&= 10^{-6} \text{ second}
\end{aligned}
$$



(SPB = 4)

$$
\text{bit time} = T_s * \text{SPB}
$$

$$
\text{bit rate} = \frac{1}{\text{bit time}} = \frac{1}{T_s * \text{SPB}} = \frac{F_s}{\text{SPB}}
$$

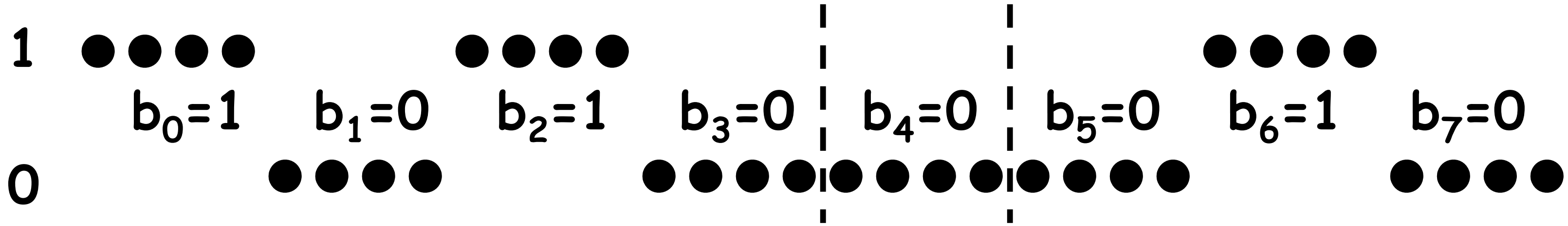
Commonly used SI (metric) prefixes

- **SI = Système international d'unités**
  (International System of Units)

Text	Symbol	Factor
tera	T	1000000000000 (10^{12})
giga	G	1000000000 (10^{9})
mega	M	1000000 (10^{6})
kilo	k	1000 (10^{3})
(none)	(none)	1
milli	m	0.001 (10^{-3})
micro	μ	0.000001 (10^{-6})
nano	n	0.000000001 (10^{-9})
pico	p	0.000000000001 (10^{-12})

Representing Bit Waveforms

- In order to describe the effect of the channel, we need a <u>convenient</u> way of representing (describing) the input.
- One way is with a graph:



- Another way is with a formula, such as:

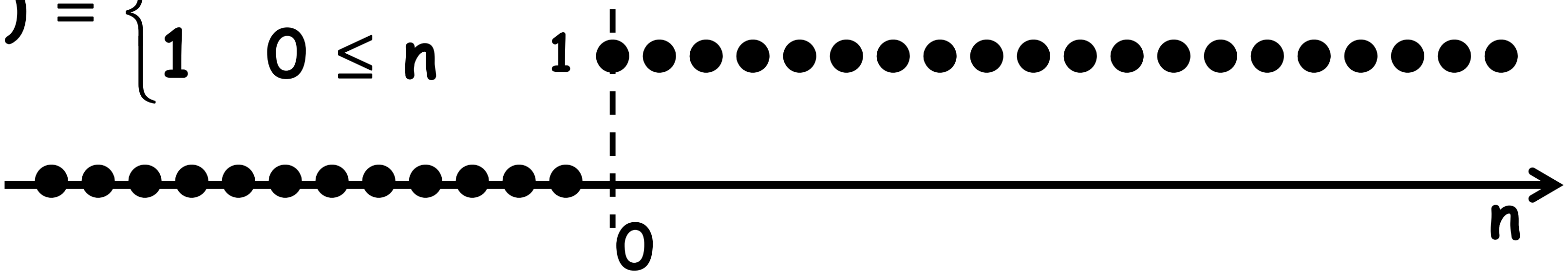
$$x(n) = \begin{cases} 1 & 0 \le n < 4 \\ 0 & 4 \le n < 8 \\ \vdots & \vdots \\ b_k & k \cdot SPB \le n < (k+1) \cdot SPB \\ \vdots & \vdots \end{cases}$$

- However, it is difficult to work with this formula. We seek a better formula.

Unit Step Function

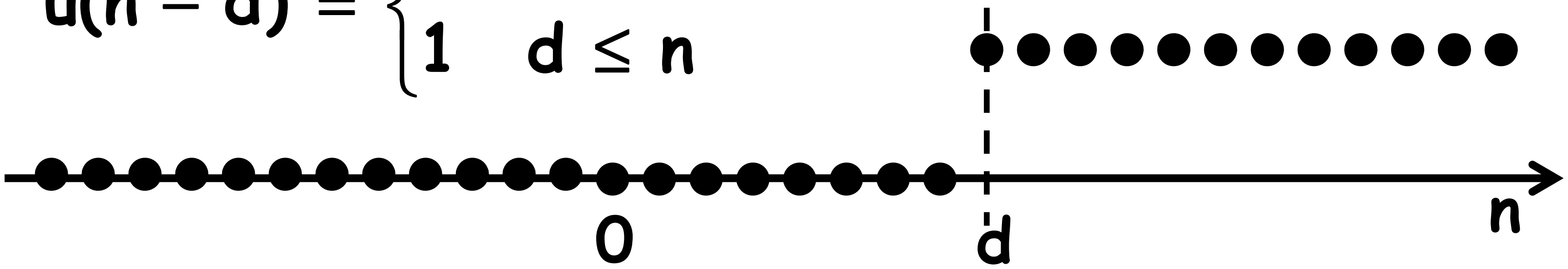
- In order to get a better formula to define a bit waveform, we define the unit step function u(n)

$$u(n) = \begin{cases} 0 & n < 0 \\ 1 & 0 \le n \end{cases}$$



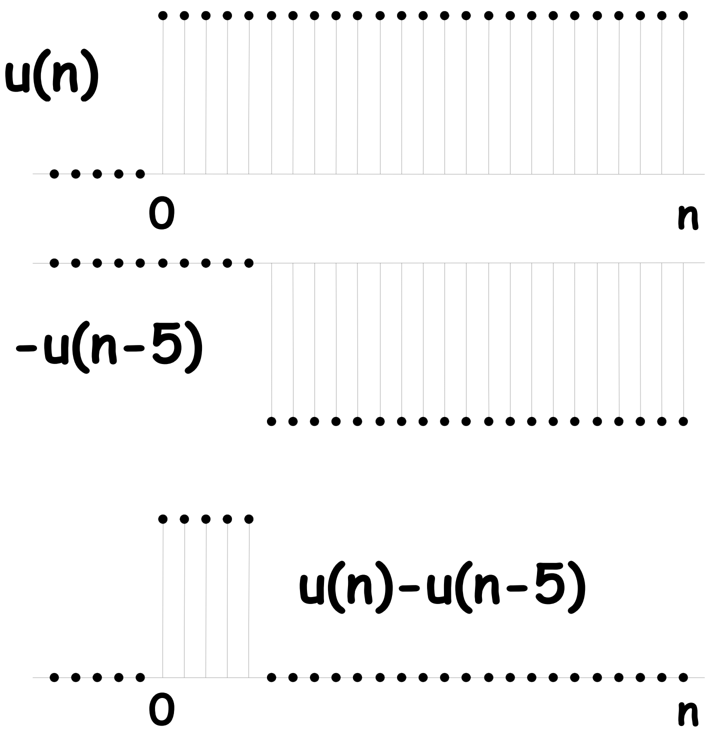
- We can delay the step as follows

$$u(n-d) = \begin{cases} 0 & n < d \\ 1 & d \le n \end{cases}$$

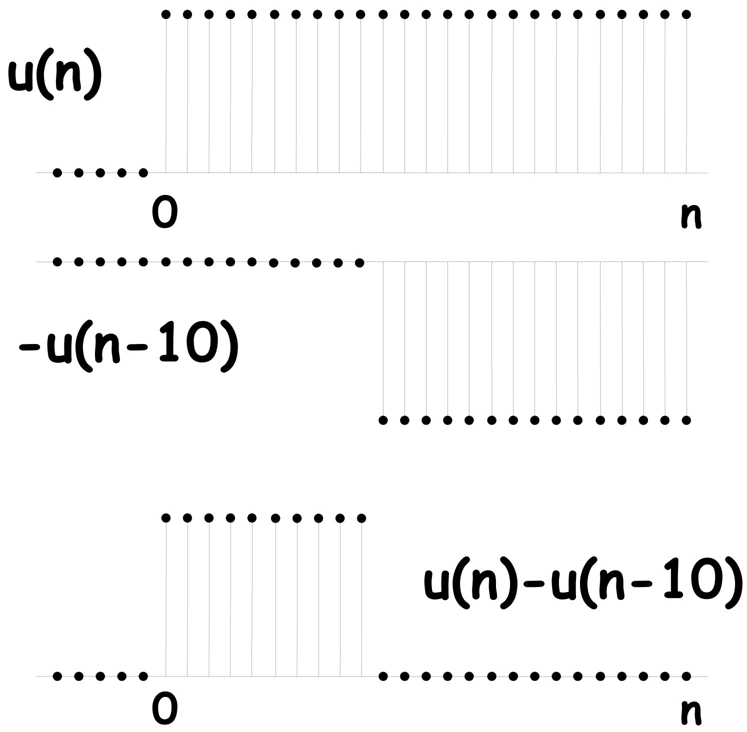


Combining Step Functions

- We can describe a single pulse as the difference between two step functions



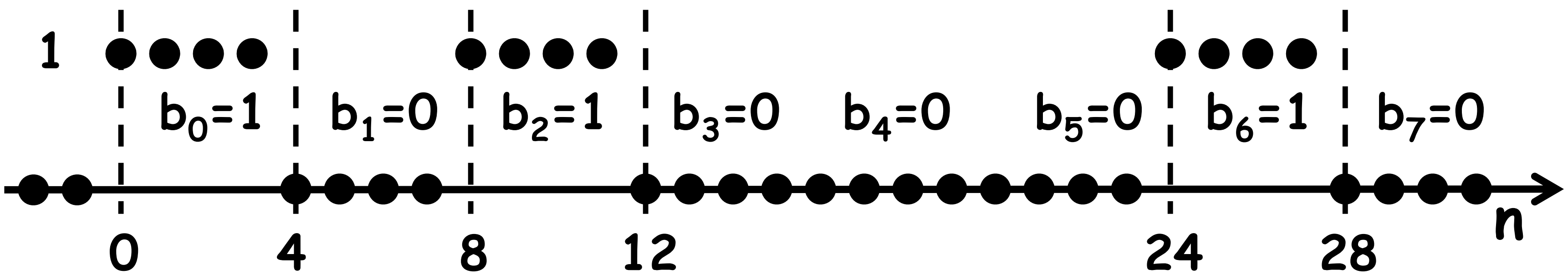
Pulse of length 5



Pulse of length 10

Representing Bit Waveforms

- We can describe any bit sequence as the sum and difference of unit step functions.
- One step function every time a bit changes
  - If the bit changes from 0 to 1 at sample D, add u(n-D)
  - If the bit changes from 1 to 0 at sample D, subtract u(n-D)
  - If there is no change, add nothing

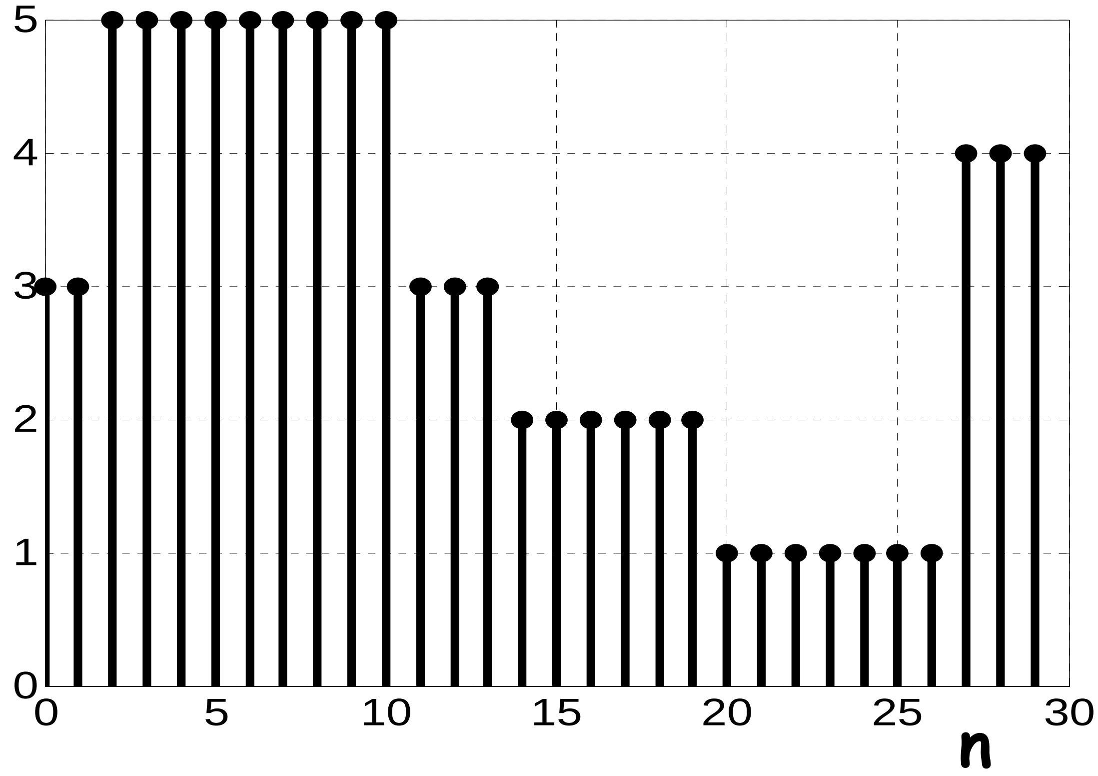


$$x(n) = u(n) - u(n-4) + u(n-8) - u(n-12) + u(n-24) - u(n-28)$$

Arbitrary Sample Waveforms

- In fact, if we allow for step functions that are scaled (multiplied by a constant) <u>any</u> sample waveform can be expressed as the sums and differences of unit step functions!

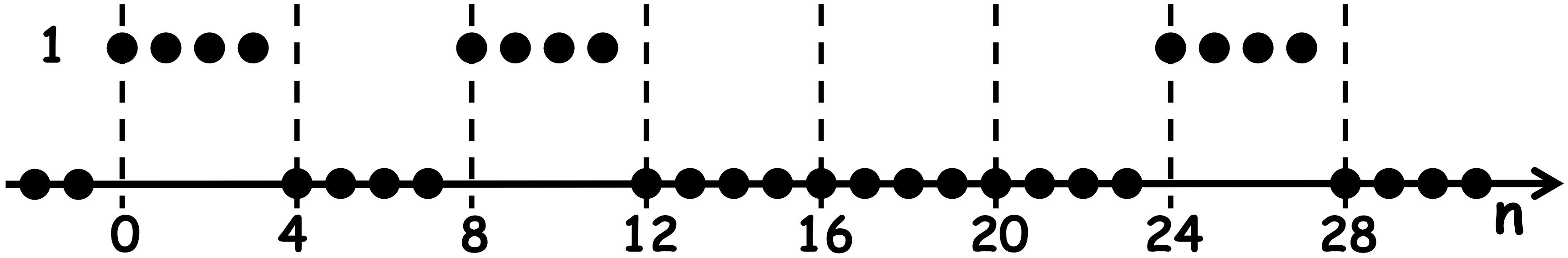
$$x(n) = 3 + 2 \cdot u(n-2) - 2 \cdot u(n-11) - u(n-14) - u(n-20) + 3 \cdot u(n-27)$$



Equivalent Representations of Waveforms

- **Verbal**: The encoding of the bit sequence 1,0,1,0,0,0,1 at 4 samples per bit.

- **Graph**



- **List, table or vector of values**

n = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 …]

x(n) = [1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 …]

- **Sum of unit step functions**

$$x(n) = u(n) - u(n-4) + u(n-8) - u(n-12)$$
$$+u(n-24) - u(n-28)$$

Uses for different representations

- The four representations are equivalent in the sense that if we know one, we can obtain any of the others. However, they are useful in different situations.
- Verbal
  - Useful for communicating between people
- Graph
  - Useful for visualizing the waveform
- List, table or vector of values
  - Useful for representation and processing inside a computer (e.g. MATLAB)
- Sum of unit step functions
  - Useful for analyzing mathematically what happens to the waveform when it passes through a communication channel.

Next time

- **We will see that we can describe the effects of transmission through a channel on <u>any signal</u> just by understanding what happens to a unit step function.**