GAA CUSTOM ELECTRONICS

From sentence to signal

Specifying waveforms for MIPS pulse sequences

Application note • Rev A

Most pulse sequences that don't work fail for one reason: the waveform was never fully specified. The syntax gets the blame, but the real gap is upstream — a description that seemed complete left something open to interpretation.

This note gives you a short checklist for pinning a waveform down, and a template you can fill in. Use it before writing a table, when asking a colleague for help, when contacting us for support, or when handing the job to an AI assistant.

The ambiguity that costs the most time

Consider a description that sounds perfectly clear:

```
Two 1 ms pulses, 10 ms apart.
```

There are two readings. The 10 ms could be measured from the leading edge of the first pulse to the leading edge of the second — or it could be the gap between them, from the falling edge of the first to the rising edge of the second. The two interpretations differ by a full millisecond, and nothing in the sentence tells you which was meant.

Now the same waveform, specified as transitions:

```
A: high at 0 ms, low at 1 ms
B: high at 10 ms, low at 11 ms
```

Nothing is left open. This is the whole idea behind the checklist: describe what changes, and when.

> **The transition model**
> A MIPS pulse sequence is a list of events, each saying *at time T*, *set output X to value V*. A waveform described as transitions maps onto that model directly, so nothing is lost in translation. Widths, gaps, and repetition rates can all be derived from the transition times — but not the other way around.

The checklist

A complete waveform specification answers these seven questions.

1. **Which outputs, and what is their idle state?** Name the channels — digital outputs A through P, DC bias channels 1 through 32, or others from the channel table — and say what each one sits at when nothing is happening (logic 0, 0 V, and so on).

2. **Where is time zero, and what starts it?** State what t = 0 refers to, and whether the sequence starts on a software command or an external trigger. If it is external, say whether every repeat needs its own trigger.

3. **What is your timing convention?** Say whether the intervals you quote are edge to edge or gaps between pulses. Better still, give each edge an explicit time and skip the question entirely.

4. **What does each pulse, level, or ramp do?** For a pulse: which output, when it goes on, when it goes off. For a level: the value it takes. For a ramp: the starting value, the ending value, and how long it takes.

5. **How does it repeat?** Describe the hierarchy from the inside out: the base pattern, then any burst around it with a count and a period, then any outer repeat with its own count and period. Say whether a count includes the first occurrence.

6. **What changes between repeats?** If something sweeps, name the parameter, the step size, the direction, the starting value, and what it advances per — per pulse, per burst, or per repeat. Say what stays fixed.

7. **What clock and units?** Give real times in ms or µs and let the table clock be chosen for you, or name the clock you want to use.

> **Rule of thumb**
> If you can draw the waveform from your own description without guessing, it is specified. If you have to make one assumption, so will whoever reads it — and they may not make the same one.

A template you can fill in

Copy this, complete it, and you have a specification that can be handed to anyone.

```
Outputs and idle states:   e.g. digital A and B idle low; DC bias 1 idle 0 V
Time zero / trigger:       e.g. t = 0 is the trigger; software, one run
Timing convention:         e.g. all times leading edge, measured from t = 0
Transitions:               e.g. A high at 0 ms, low at 1 ms
                                B high at 10 ms, low at 11 ms
Repetition:                inner  count ____  period ____
                           outer  count ____  period ____
What sweeps:               parameter ____  step ____  per ____  from ____
                           fixed ____
Clock:                     e.g. 656250 Hz internal, or leave to be chosen
```

Worked example 1 — a digital pulse train

The specification:

```
Outputs:      digital B, idle low
Time zero:    software trigger
Convention:   times are from t = 0, leading edge
Transitions:  B high at 152 µs, low at 457 µs
Repetition:   repeat 10 times, period 610 µs
Sweep:        none
Clock:        656250 Hz
```

The resulting sequence, with counts in clock cycles at 656250 Hz (656 clocks ≈ 1 ms):

```
STBLDAT;0:[A:10,100:B:1,300:B:0,400:];
```

Every field traces back to a line of the specification: the loop repeats 10 times, `100:B:1` and `300:B:0` are the two transitions, and the loop length `400` is the period.

Worked example 2 — a two-segment ramp

The specification:

```
Outputs:      DC bias channel 1, idle 0 V
Time zero:    software trigger
Convention:   segment durations, run back to back
Ramp:         0 V to 100 V over 50 ms
              hold 100 V for 25 ms
              100 V to 0 V over 25 ms
Repetition:   repeat 3 times
Sweep:        none
Clock:        656250 Hz
```

The resulting sequence:

```
STBLDAT;0:[A:3,0:65:0:[P:100,0:<:50:129:2:>:75:129:-4,656:],100:193:0];
```

Each field of this sequence is explained in full in the pulse sequence examples note. Conditional ramping must be enabled with STBLVDLT,TRUE before download.

Ambiguities worth naming

Phrase	What to say instead
"10 ms apart"	Say whether it is edge to edge or the gap between pulses, or give both edges explicit times.
"3 pulses"	Say whether the count includes the first pulse. Three pulses and three repeats after the first are different waveforms.
"repeat every 100 ms"	Confirm 100 ms is the period from one start to the next, not the idle time between.
"ramp to 100 V"	Give the start value, the end value, and the duration. A ramp rate without a starting point is incomplete.

Using this with an AI assistant

The same completed template works as a prompt. Provide the MIPS pulse sequence reference manual along with it, so the assistant works from the actual command set rather than from memory, and ask it to explain each field of the table it produces. Treat the result as a starting point: a good assistant will tell you which behavior it could not confirm from the documentation, and that is exactly the part to check first.

Always verify on hardware

Any generated sequence — written by you, by us, or by an assistant — should be verified on an oscilloscope before use in an experiment. Confirm amplitudes and timing against your specification, and start with conservative values.

Related documents

- **MIPS Pulse Sequence Generation** — reference manual: event syntax, channel table, table commands.
- **MIPS Pulse Sequence Generation, Application Note** — worked examples, from a single pulse to multi-segment ramps.