DTP2 Code Checkliste

Allgemein	**Kommentar**
Nur synchrone Prozesse	Keine Logik auf das Clock-Signal
Reset ist asynchrones Signal und darf nur im getakteten Prozess verwendet werden	Oder Synchrones Reset, aber gleich für ganzes Projekt (clock und reset strategy kurz kommentieren)
Keine Latches (keine asynchronen Speicher)	Auch keine Komb-Loops, und keine offene Eingänge
In kombinatorischen Prozessen müssen alle Eingangssignale auf in die sensitivity Liste (ausser der code ist für VHDL 2008 geschrieben)	Mit VHDL-2008 muss nur «all» in die Sensitivity liste eingetragen werden
Signale aus unterschiedlichen Clock Domains (oder asynchrone Eingänge) müssen synchronisiert werden	Auch das globale reset Signal
Bevorzugt Moore, wenn möglich keine Mealy Maschinen bauen	Das bedeutet Eingänge gehen durch die FSM Ansteuerlogik (Eingangslogik), dann durch FFs, und bestimmen nicht direkt die Ausgänge. Gute Praxis ist separate Prozesse für Ansteuerlogik, FF und Ausgangslogik.
Top-Testbench (TB) pro Milesteine (oder nur eigenen testcase.dat für jeden Meilenstein), oder Block-TB wo sinnvoll. Die testcases decken verschiedene Betriebsarten und kritische Punkte ab	z.B. Einfluss von Eingangssignalen und Überlauf von Zähler.
Kurz Erklärung von Testszenario für Testbenches.	Falls self-checking TB nicht möglich, Kommentare über Testszenario und visuelle checks.
Mind. eine TB / Design /Integr. pro Person	Jede hat alle Tätigkeiten ausprobiert
VHDL Style	**Kommentar**
VHDL Code muss leserlich formatiert sein. (Klammern, IF-Else, Begin, End)	Benutzen Sie im emacs VHDL Beautify (ctrl-c dann b)
Klar getrennte Prozesse	Getrennte getaktete und kombinatorische Prozesse. Wo sinnvoll: getrennte komb. Prozesse für Eingangslogik/Ausgangslogik
Sinnvolle Anwendung von CASE vs IF-ELSE	Check ob Logik prio oder nicht braucht Aber nicht übertreiben: CASE/WHEN mit nur 1-bit nicht sinnvoll, besser IF-ELSE
Nutzung der Prioritäten von IF-ELSE um übersichtlichen Code zu gestalten	z.B. wenn in der Logik keine Ausgabe gemacht wird, solange enable_xxx nicht wahr ist
Keine Funktionalität in der Strukturellen Entity (Top Level)	Hierarchische Blöcke nur mit Komponenten und Instanziierungen Vermeidung RTL-Design und Hierarchie zu mischen. Kurzen Kommentar abgeben falls dies nicht möglich ist.
Sinnvolle Aufteilung von logischen Prozessen	Packt verwandte Funktionalität, und behält Überblick / Komplexität im Griff für Code-Leser
Entity Ports haben std_logic oder std_logic_vector als Daten-Typ	Ausnahmen (wo wirklich sinnvoll, e.g. mid_array) kommentieren
Keine Signal forcing	Nur eine Ausnahme erwartet (clock_divider). Kommentieren warum.
Code Dokumentation	**Kommentar**
Header mit Namen des Erstellers und Änderungsliste. Kurze Beschreibung was der Block macht	Falls Sie ein Version-Management Tool benutzt haben, kann man sich in der Änderungsliste darauf beziehen.

Sinnvolle Kommentare	An Code-Stellen die nicht naheliegend sind (e.g. support running status MIDI)
Sinnvolle Signalnamen	Schlecht nur Name „State“ für FSM Codierung
Signalnamen müssen Polarität anzeigen	z.B. reset_n falls low aktiv
Naming convention for ports, signals and process	Uniform along the project for all blocks
Synthese	**Kommentar**
Warnungen	Bleibende Warnung checken und kommentieren welche akzeptabel sind
Check Resource Utilization	FFs, Komb-Log, Mem-Blocks