

Program nauczania sponsorowany przez firmę INTEL



Advanced ASIC Design

Instrukcja Laboratoryjna

Ćwiczenie nr 1: SPYGLASS LINT

I. Zapoznanie z oprogramowaniem

1. Uruchomienie narzędzia SpyGlass

Do uruchomienia narzędzi Synopsys (a więc również oprogramowania SpyGlass) niezbędne jest ustawienie odpowiednich zmiennych i ścieżek systemowych. Zmienne te można ustawić poprzez uruchomienie przygotowanego wcześniej skryptu wywołując komendę:

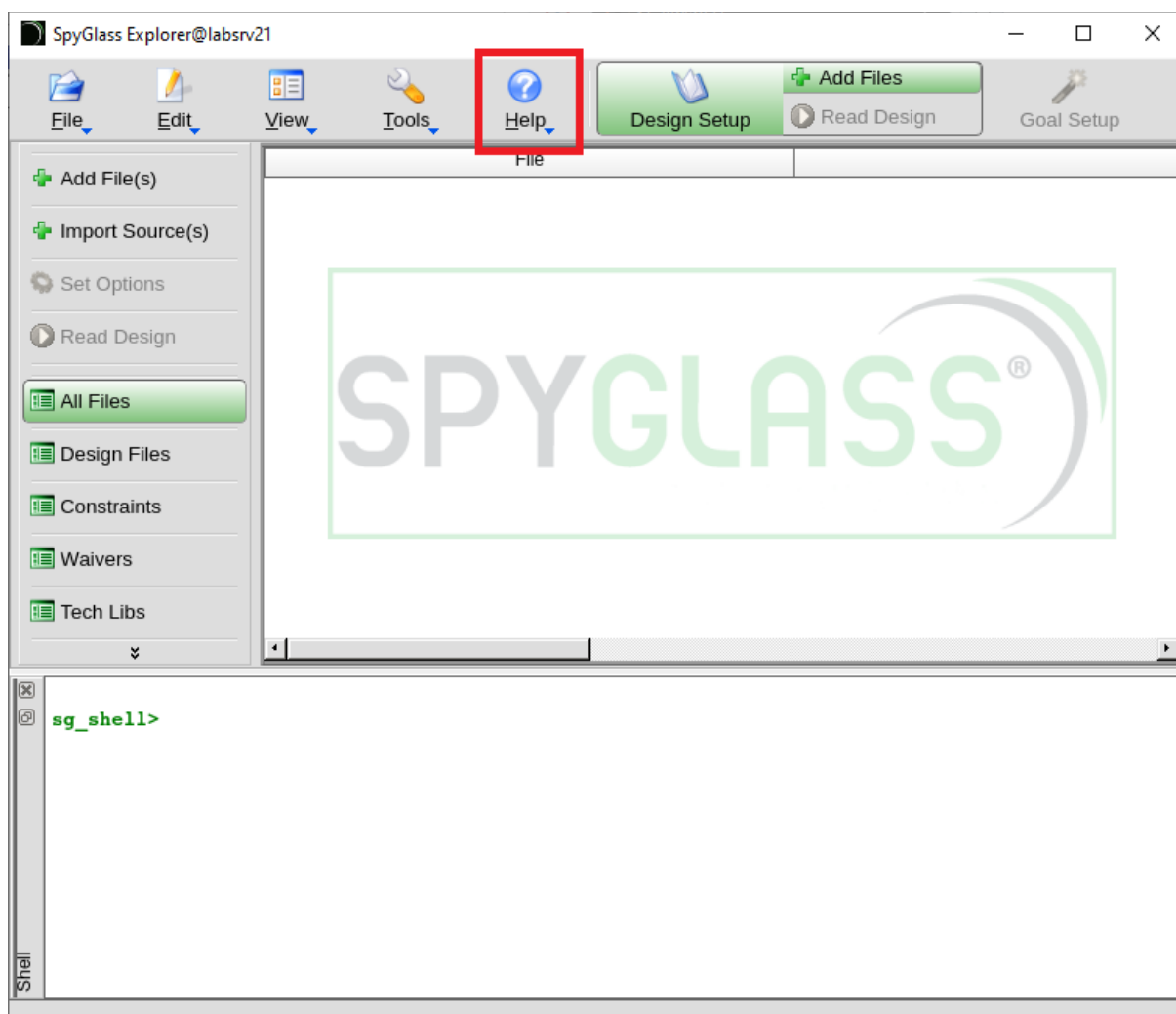
```
add-synopsys-FEV-all
```

Następnie należy przejść do swojego katalogu roboczego, gdzie można już uruchomić narzędzie SpyGlass poprzez komendę:

```
spyglass
```

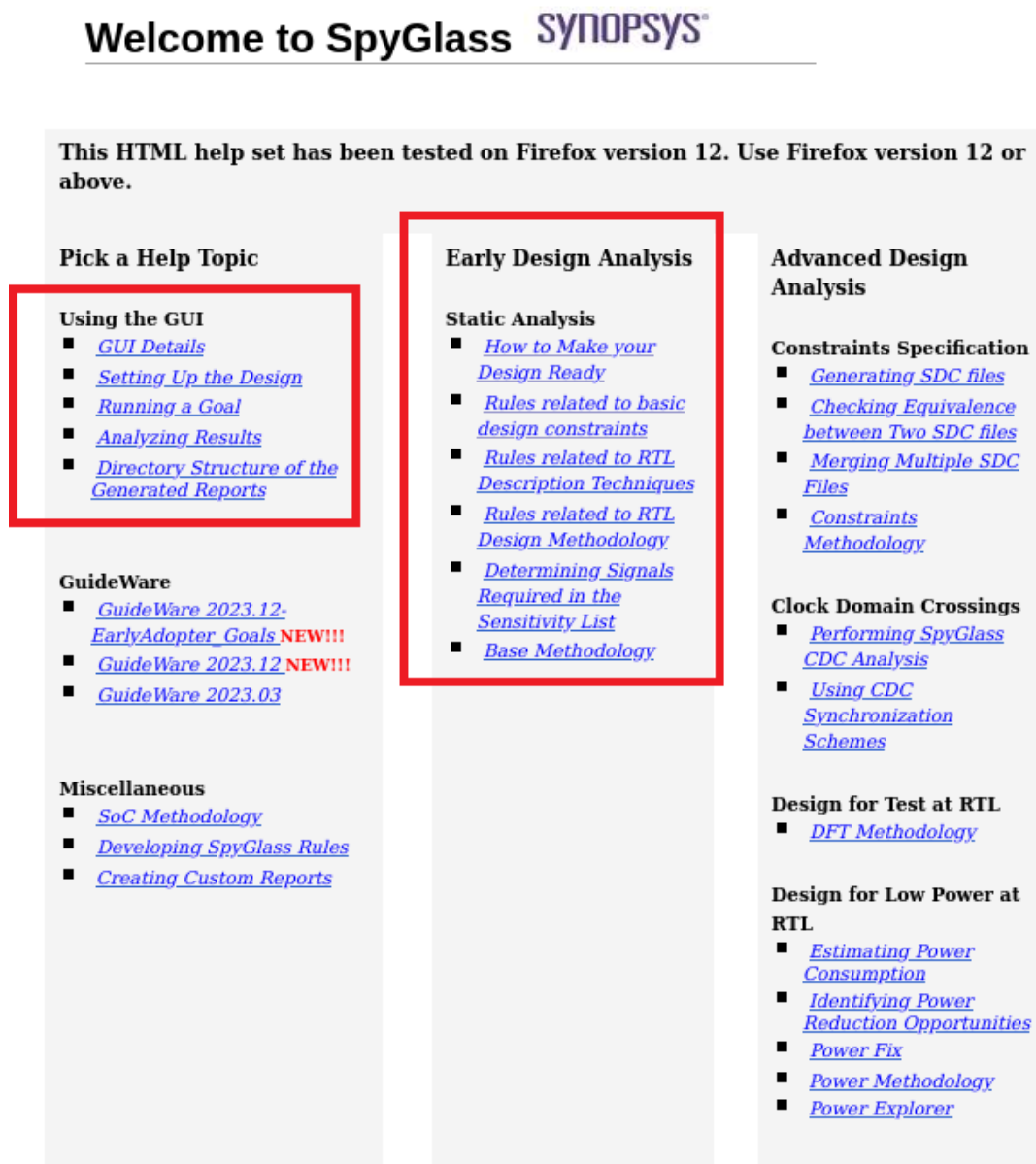
2. Zapoznanie się z narzędziem SpyGlass

Po wywołaniu komendy zostanie otwarty graficzny interfejs użytkownika:



Rys. 1 Główne okno narzędzia SpyGlass

Zapoznanie z oprogramowaniem najlepiej rozpocząć od uruchomienia wbudowanego narzędzia pomocy, poprzez kliknięcie na ikonę Pomocy (*Help -> SpyGlass Help*):

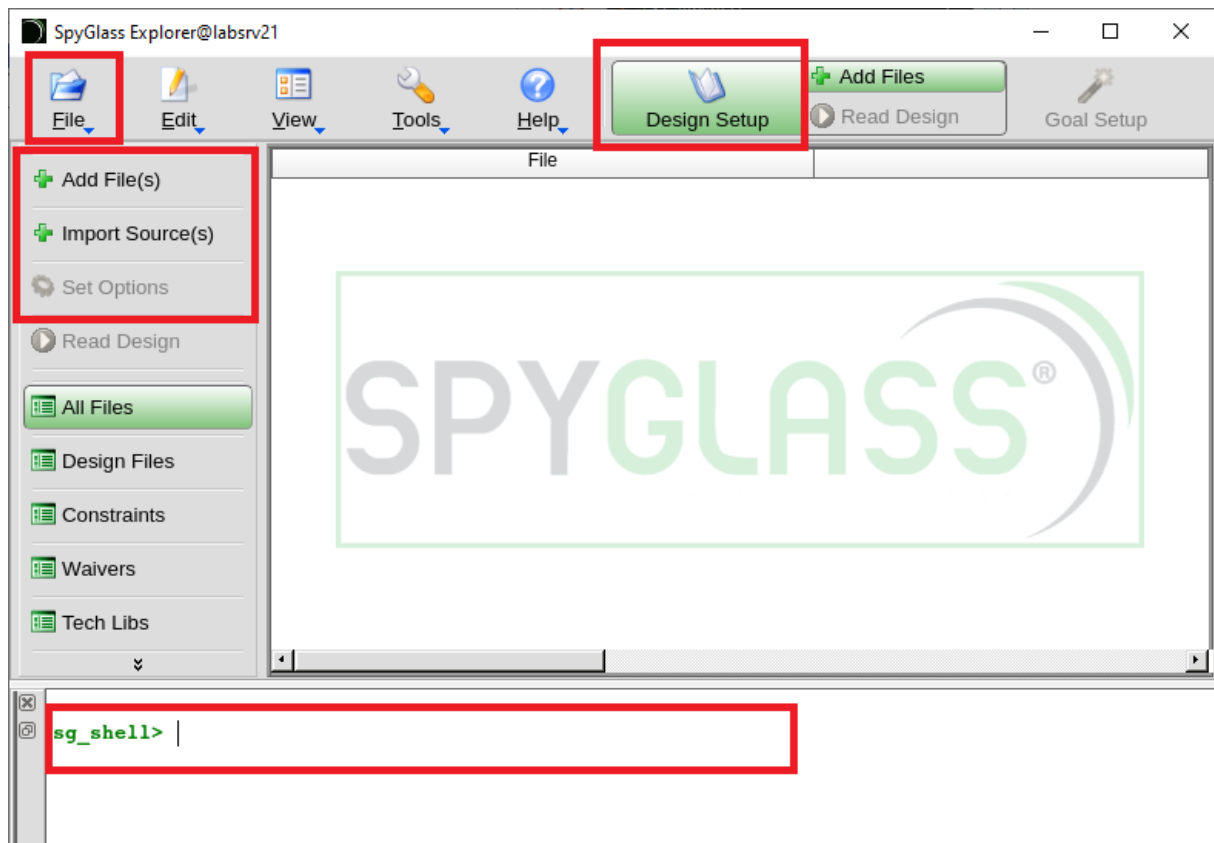


Rys. 2 SpyGlass - Pomoc

W pomocy można znaleźć między innymi poradnik obsługi interfejsu graficznego (*Using the GUI*) oraz opis podstawowej analizy kodu języku opisu sprzętu (*Static Analysis*) wraz z opisem niektórych reguł sprawdzanych podczas „Linting’u” kodu. Pozostałe reguły można znaleźć w innych rozdziałach pomocy (*SpyGlass Documentation -> Produkt Dokumentation -> Lint -> Rules in SpyGlass lint*).

3. Wczytywanie plików

Pliki do oprogramowania SpyGlass można wczytywać na kilka sposobów. Pierwszym z nich jest użycie interfejsu graficznego:



Rys. 3 Instrukcja dodawania plików

Następnie należy przejść do menu *Design Setup* i użyć przycisków *Add File(s)* lub *Import Sources(s)*. W pierwszym przypadku wszystkie pliki należy dodawać pojedynczo wskazując ich lokalizację. W drugim przypadku wystarczy wskazać jeden plik o rozszerzeniu „*.f”. Plik ten powinien zawierać ścieżki i nazwy do wszystkich plików używanych w danym projekcie, dzięki czemu wszystkie pliki wczytają się automatycznie.

Pliki można również wczytywać za pomocą komend konsoli SpyGlass:

```
read_file -type verilog "ścieżka_i_nazwa_pliku.v"
```

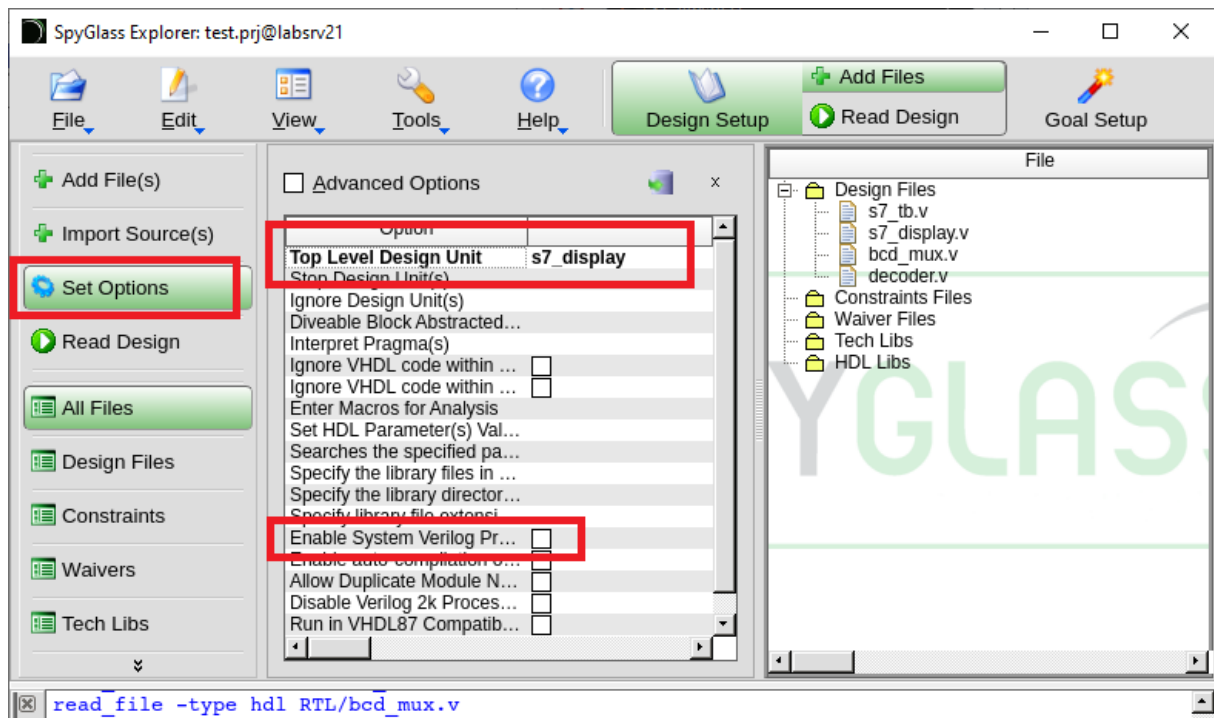
```
read_file -type vhdl "ścieżka_i_nazwa_pliku.vhdl"
```

```
read_file -type sourcelist "ścieżka_i_nazwa_pliku.f"
```

W każdym momencie swoją pracę można zapisać poprzez zapisanie projektu, przez menu *File -> Save Project As*. Po zapisaniu projektu można go otworzyć poprzez graficzny interfejs *File -> Open Project* lub uruchamiając program z odpowiednim przełącznikiem:

```
spyglass -project "ścieżka_i_nazwa_projektu.prj"
```

Po dodaniu plików, zostanie uaktywniona graficzna opcja *Set Options* (oznaczona kołem zębatym), pozwalająca między innymi na ustawienia który z dodanych modułów znajdujących się w plikach jest modułem nadrzędnym (Top Level Design Unit):



Rys. 4 Opcje projektu

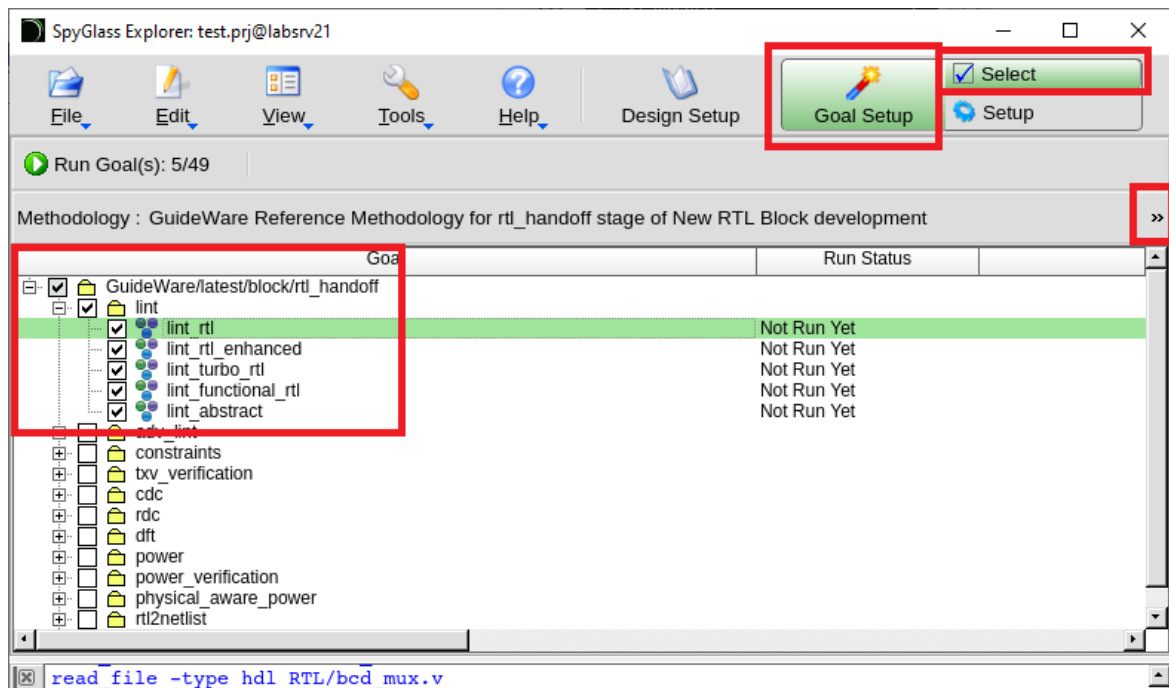
Moduł nadrzędny można także ustawić komendą konsoli SpyGlass:

```
set_option top „nazwa_modułu”
```

Jeżeli w projekcie używamy modułów napisanych w języku System Verilog, w menu „Set Options” należy zaznaczyć opcję *Enable System Verilog Processing*.

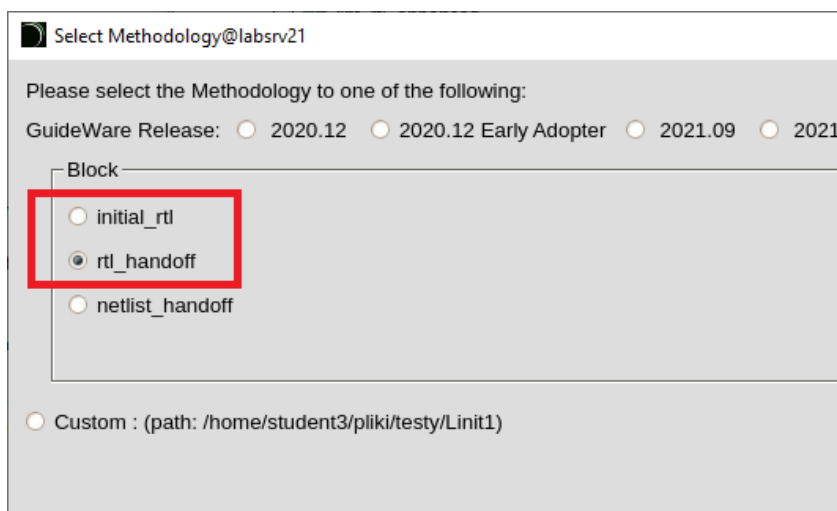
4. Wybieranie metodologii celów oraz zasad

Narzędzie SpyGlas zawiera bardzo dużą grupę zasad sprawdzania kodu. Nie wszystkie z nich są włączone automatycznie podczas uruchamiania programu. Dla porządku zostały one podzielone na metodologie, kategorie oraz scenariusze. Aktywny scenariusz zasad wybieramy w menu *Goal Setup* w zakładce *Select*.



Rys. 5 Wybór Reguł

Pierwszą nadrzędną grupą, którą wybieramy jest Metodologia (Methodology). Wybieramy ją poprzez wciśnięciem symbolu „>>”. Dostępne są metodologie GuideWare, o których informację można znaleźć w pomocy narzędzia SpyGlass.

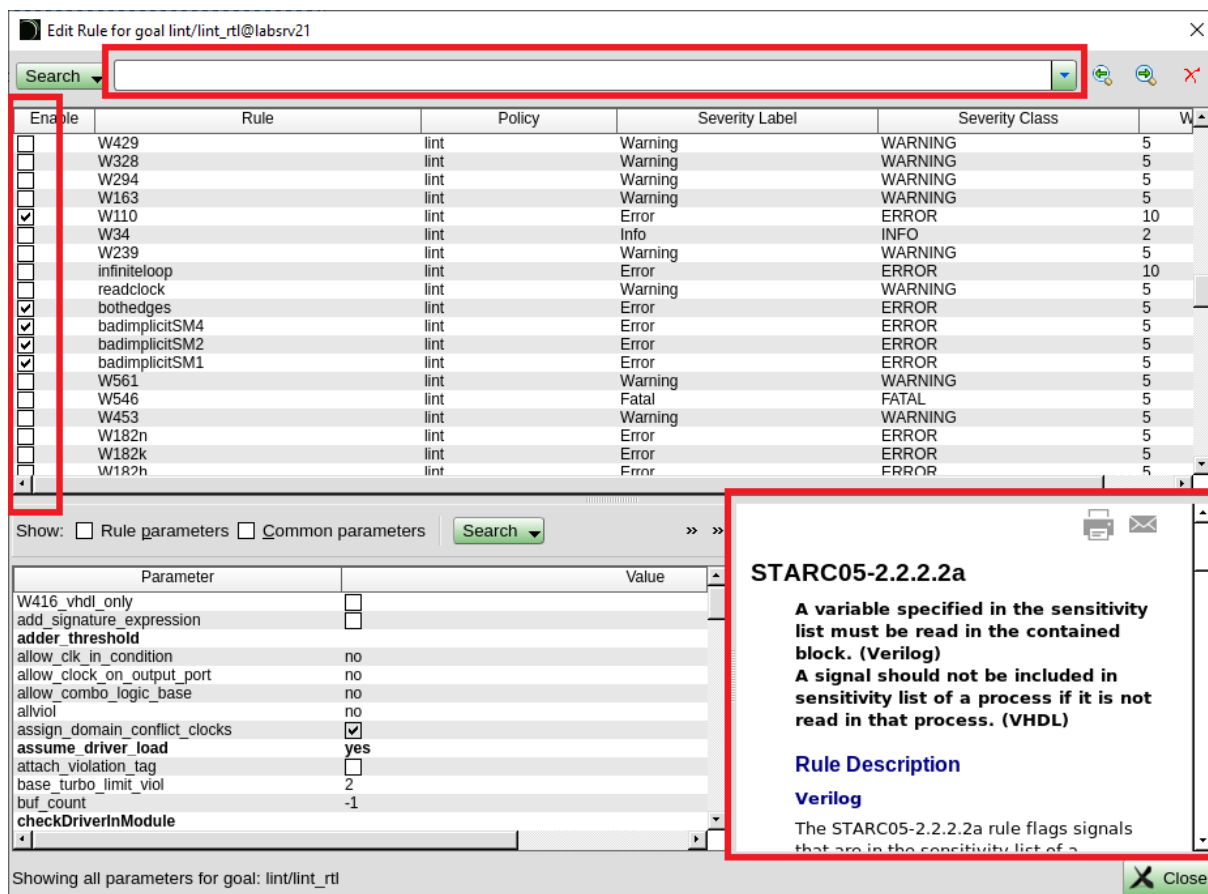


Rys. 6 Wybór Metodologii

Na tych zajęciach zajmować będziemy się GuideWare Release 2023.12 oraz grupami Block: *initial_rtl* oraz *rtl_handoff*.

Po wybraniu metodologii, możemy przejść do wyboru poszczególnych grup zasad (Rys. 5). Na dzisiejszych zajęciach będziemy używać grupy „*lint*”. Zaznaczmy tutaj grupy i scenariusze, których zasady powinny być aktywne podczas następnego sprawdzania kodu.

Klikając prawym klawiszem myszy na poszczególne scenariusze (np. w grupie lint) możemy uruchomić opcję *Edit rules and Parameters*.



Rys. 7 - Edytowanie zasad

Uruchomione menu pozwala między innymi na obejrzenie wszystkich dostępnych zasad w obrębie metodologii. Nie wszystkie z nich domyślnie są włączone. W oknie tym możemy również włączać i wyłączać aktywne zasady, edytować ich parametry oraz wyszukiwać poszczególne zasady. Menu w prawym dolnym rogu okna, przekierowuje nas do pomocy, gdzie każda z zaznaczonych reguł jest opisana.

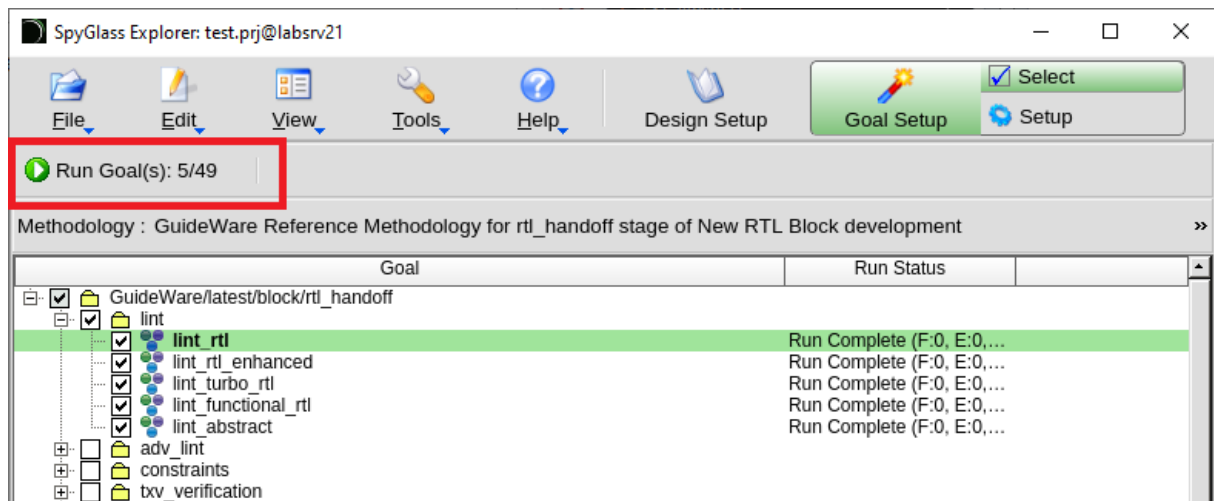
Zasady możemy też dodawać i usuwać poprzez komendy konsoli SpyGlass:

```
set_goal_option addrule „nazwa_reguły”
```

```
set_goal_option ignorerule „nazwa_reguły”
```

5. Analiza Kodu

Po dodaniu plików zawierających kod oraz ustawieniu reguł, możliwe jest uruchomienie narzędzia sprawdzania kodu. Narzędzie te uruchamiamy poprzez wciśnięcie przycisku *Run Goal(s)* w zakładce *Goal Setup*:

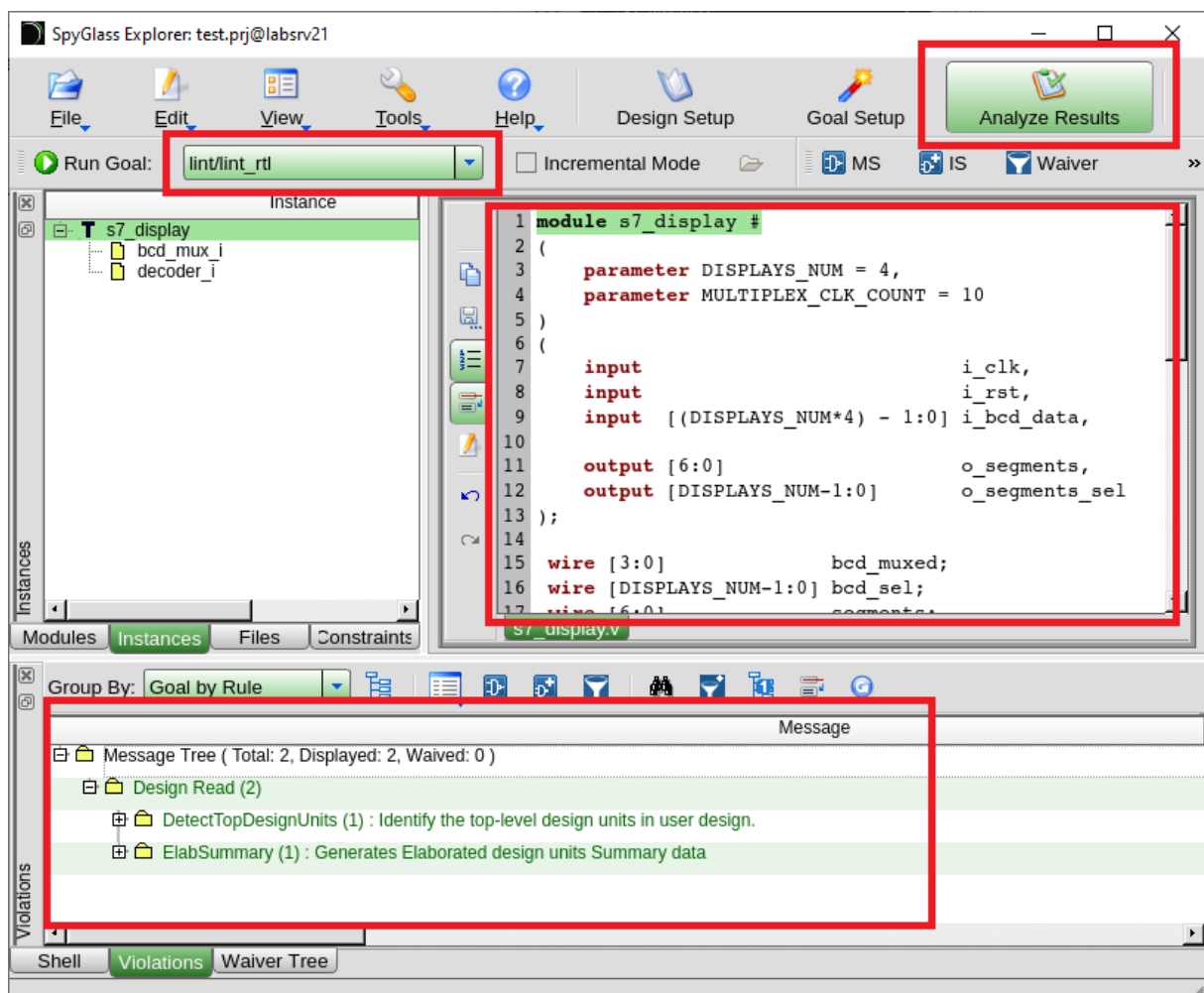


Rys. 8 Uruchomienie analizy kodu

Lub komendę konsoli SpyGlass:

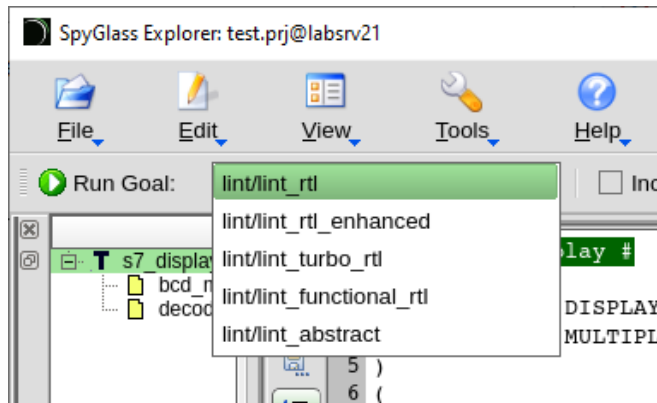
run_goal

Po przejściu do zakładki *Analyze Results*, dostępne będzie okno wyświetlające wszystkie naruszenia wybranych reguł.



Rys. 9 Analiza Kodu

Naruszenia zgłaszane będą na dole ekranu w zakładce „Violations”, ale będą one wyświetlane tylko dla aktywnego scenariusza. Zmienić aktualny scenariusz można poprzez listę obok przycisku „Run Goal”:



Rys. 10 Zmiana aktywnego scenariusza

Należy więc sprawdzić naruszenia dla każdego scenariusza z tej listy.

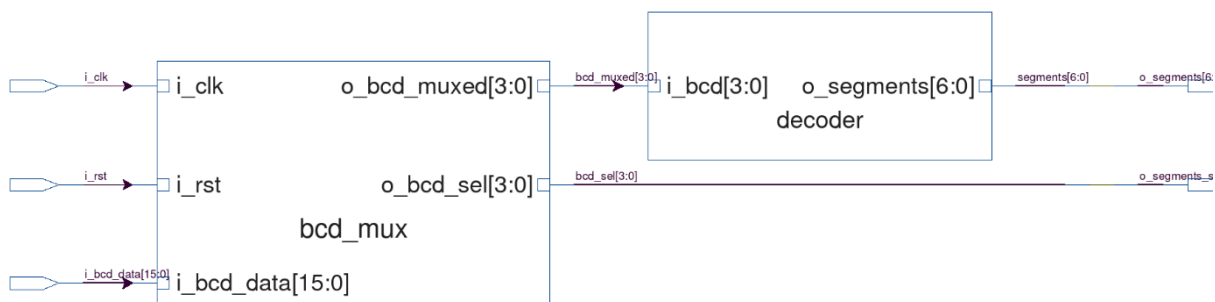
Podczas usuwania naruszeń kodu, warto otworzyć sobie okno pomocy poprzez menu „View -> Windows -> Help Viewer”. Po jego uruchomieniu będzie tam wyświetlana pomoc dotycząca aktualnego naruszenia.

II. Przebieg Ćwiczenia:

1. Uruchomić narzędzie SpyGlass i zapoznać się z jego interfejsem.
2. Wczytać w dowolny sposób kod układu multipleksowania 4 cyfrowego wyświetlacza siedmiosegmentowego ze wspólną anodą, dostarczonego razem z instrukcją:

Kod składa się z 4 plików znajdujących się w katalogu RTL. Plik nadrzędny (Top module) projektu to plik „s7_display.v” i to on powinien być ustawiony jako plik nadrzędny w oprogramowaniu SpyGlass. Plik „s7_tb.v” zawiera testbench pozwalający na uruchomienie symulacji.

Dostarczone pliki zawierają również plik „filelist.f” zawierających listę plików opisujących układ sterowania wyświetlacza oraz skrypt „runsym.sh” pozwalający na uruchomienie symulacji w środowisku „Verdi” po uruchomieniu komendy „sh runsym.sh” w konsoli systemu linux.



Rys. 11 Schemat modułów wyświetlacza

Dostarczony projekt składa się z 2 modułów. Modułu „bcd_mux”, który odpowiedzialny jest za wybieranie aktualnie multipleksowanej cyfry w kodzie BCD oraz wysterowanie sygnału anody. Moduł „decoder” to układ kombinacyjny dekodera, który dekoduje kod BCD na kod wyświetlacza siedmiosegmentowego. Moduł „s7_display” to moduł nadrzędny łączący oba te moduły.

W testbenchu (plik „s7_tb.v”) została dodana funkcja dekodująca kod wyświetlacza siedmiosegmentowego na kody ASCII. Pozwala ona podczas symulacji wyświetlać kody ASCII poszczególnych cyfr pozwalając na prostą weryfikację pracy układu.

3. Dostarczone pliki opisujące RTL zawierają błędy, które należy usunąć.

Używając oprogramowania SpyGlass, należy znaleźć wszystkie naruszenia reguł wykryte przez to oprogramowanie i je usunąć. **W sprawozdaniu wypisać nazwy naruszonych reguł i opisać niezbędne zmiany w kodzie.**

Podczas sprawdzania reguł należy mieć włączoną metodologię: *GuideWare Release 2023.12, Block, initial_rtl* oraz *rtl_handoff*, oraz wszystkie scenariusze z katalogu *lint*.

4. Po usunięciu błędów, należy uruchomić symulację i zweryfikować pracę układu.

W sprawozdaniu umieścić screen przedstawiający przebiegi czasowe pokazujące poprawną pracę układu. Czy jakieś błędy powodujące błędne działanie układu nie zostały zgłoszone?

5. Wybrać i uruchomić dodatkowe reguły

Wybrać 5 reguł z “Severity Label: Guideline” które będą naruszone w aktualnej wersji kodu (np. PortComment, FunctionComment, ClkName). Następnie wprowadzić zmiany w kodzie zawarte w tych regułach.

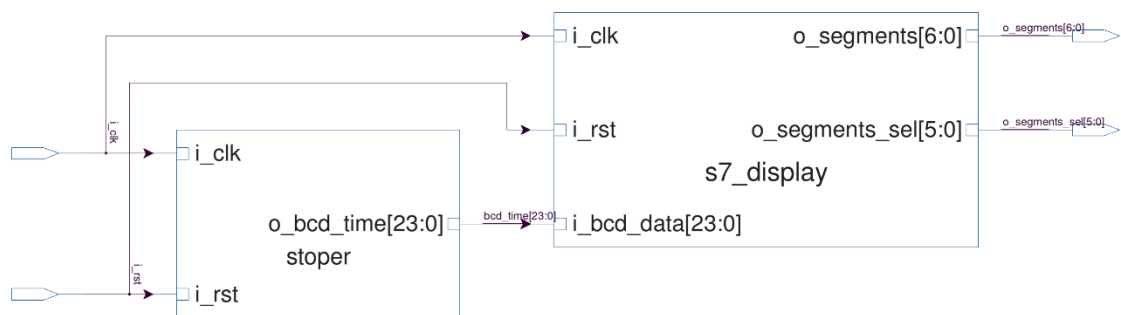
W sprawozdaniu wypisać nazwy naruszonych reguł i opisać niezbędne zmiany w kodzie. Opisz, czy twoim zdaniem uruchamianie takich reguł jest potrzebne?

6. Zaprojektować układ stopera, odliczającego milisekundy, sekundy oraz minuty.

Układ powinien odliczać czas od dezaktywacji sygnału reset. Jego wyjścia powinny umożliwiać wyświetlenie czasu na 6 multipleksowanych wyświetlaczach siedmiosegmentowych. Trzy cyfry na milisekundy, dwie na sekundy oraz jedna na minuty. Wykorzystać moduły i testbench wykorzystywany w poprzednich ćwiczeniach.

Zaprojektowany moduł powinien znajdować się w pliku stoper.v. Moduł ten powinien mieć dwa wejścia *i_clk* oraz *i_rst* oraz jedno wyjście *o_bcd_time* o szerokości 24bitów (6 cyfr w kodzie bcd).

Należy również dodać i zaprojektować nadrzędny moduł *s7_stoper* (plik *s7_stoper.v*) w którym połączone zostaną moduły zliczania czasu (*stoper*) oraz wyświetlania (*s7_display*). Następnie należy wprowadzić odpowiednie zmiany do testbenchu umożliwiające uruchomienie symulacji i zweryfikowanie pracy układu.



Rys. 12 Schemat modułów projektowanego stopera

W sprawozdaniu zamieścić kod układu zliczającego czas (stoper.v) oraz wypisać reguły, które zostały naruszone podczas projektowania tego modułu.