

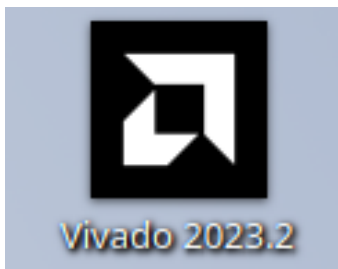


Laboratorní cvičení z předmětu CITE

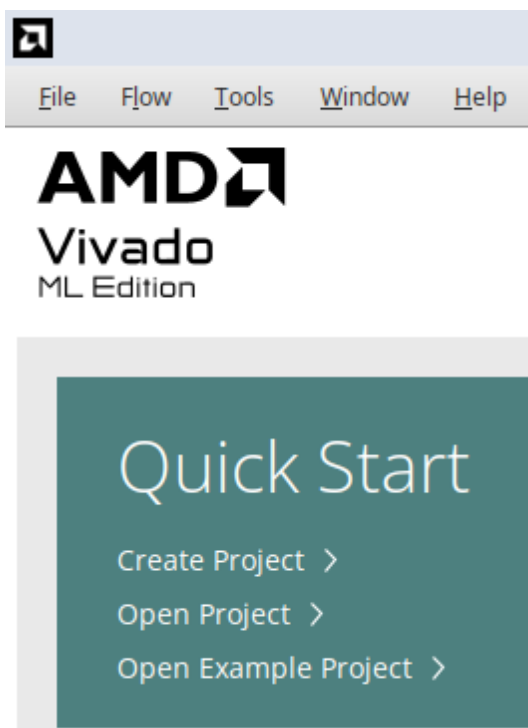
Stavové automaty



1. Nejprve se seznámte s popisem stavových automatů pomocí orientovaných grafů v seminární části cvičení.
2. Z kurzu předmětu na elearning.tul.cz stáhněte a rozbalte projekt LAB09.zip.
3. *Cesta k projektu nesmí obsahovat diakritiku, mezery nebo speciální znaky. Vhodné umístění je v laboratoři A107 vaše domovská složka, ve kterém si můžete vytvořit podadresář s vaším jménem bez diakritiky.*
4. Otevřete Vivado – dvakrát klikněte na spouštěč na ploše:



- 5.
6. V úvodním okně aplikace klikněte na Quick Start – Open Project:

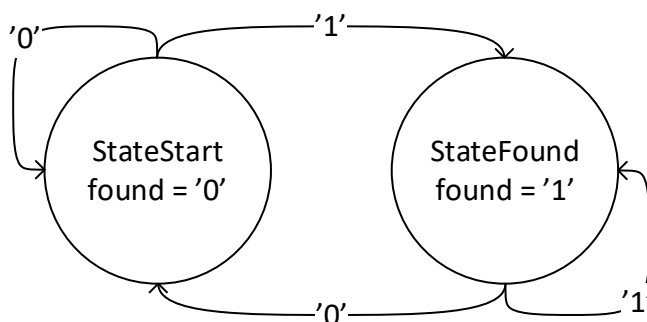


- 7.
8. Otevřete soubor LAB09.xpr uvnitř dekomprimované složky.



Detektor sekvence pomocí stavového automatu typu Moore

1. Otevřete soubor **fsm_moore.vhd**.
2. V souboru se nachází popis dvouprocesový popis stavového automatu typu Moore.



- a. Ke kódování stavů využíváme datový typ **enum** definovaný v deklarační části architektury. Obecný a konkrétní zápis deklarace je následující:

```
type enum_type_name is (Enum0 [, Enum1, ...]);
```

```
type fsm_type is (StateStart, StateFound);
```

- b. Signál **current_state** typu **fsm_type** je použit jako registr – paměť stavového automatu. Signál **next_state** realizuje výstup kombinační logiky přechodové funkce automatu.

```
signal current_state, next_state : fsm_type;
```

- c. Celý automat je kódován ve dvojici procesů.
 - i. Paměťová část automatu je popsána v procesu **mem**, přechodová funkce je popsána příkazem **case** v kombinačním procesu **transfer**
 - d. Do sešitu zakreslete graf automatu podle popisu v kódu.
3. Upravte kód automatu tak, aby detekoval sekvenci – čtyřbitové binárně zapsané číslo vašeho pracoviště v nekonečném vstupním proudu bitů na vstupu **seq**.
 - a. Nejprve nakreslete graf automatu (výstupy v ohodnocených uzlech)
 - b. Budete muset upravit definici typu **fsm_type** a proces **transfer**



- c. Upravte i přechodovou a výstupní funkci v klauzuli others – stavy hřbitova a rajská zahrada.

Mealyho automat

1. Nakreslete orientovaný graf automatu úlohy v předchozím bodu, ale tentokrát typu Mealy. Výstupní funkce se nachází na ohodnocených přechodech.
2. Otevřete soubor **fsm_mealy.vhd**
3. Upravte kód automatu tak, aby realizoval zadanou funkci.
4. Porovnejte chování automatů v simulaci **fsm_compare**
 - a. Upravte kód tak, aby došlo jak k nalezení hledané sekvence tak, aby byl otestován alespoň jeden případ, kdy sekvence detekována není.

Příklady na procvičení

Úlohu nejprve popište grafem. Realizaci proveďte pomocí automatu Mealy nebo Moore. Automat otestujte v simulátoru.

1) Navrhněte synchronní sekvenční obvod se dvěma vstupy $[x_1, x_2]$ a s výstupem $[z]$. Obvod porovnává dvě sériově vstupující slova. Když se na vstupech zároveň objeví shodné posloupnosti o délce alespoň 3 bitů, výstup z se nastaví do log. 1. Výstup se navrátí do nuly zároveň s první rozdílnou dvojicí bitů na vstupech.

2) Navrhněte synchronní sekvenční obvod s třemi vstupy $[x_1, x_2, x_3]$ a s výstupem $[z]$. Vstupy tvoří v jednom časovém okamžiku 3bitový vektor. Obvod porovnává tyto vektory. Výstup $z = 1$ pouze tehdy, když se na vstupech objeví posloupnost vektorů s více než jednou jedničkou alespoň ve třech za sebou generovaných vektorech. Výstup z se navrátí do nuly v tom okamžiku, kdy tato podmínka pomine. Řešte tak, že ve zvláštním kombinačním obvodu vyhodnotíte počet jedniček ve vstupním vektoru, výstupní jednobitovou veličinu využijte jako vstup navazujícího sekvenčního obvodu.

3) Navrhněte synchronní sekvenční obvod s čtyřmi vstupy $[x_0, x_1, x_2, x_3]$ a s výstupem $[z]$. Vstupy tvoří v jednom časovém okamžiku 4bitový vektor. Obvod porovnává tyto vektory. Výstup $z = 1$ pouze tehdy, když se na vstupech objeví posloupnost vektorů s lichou paritou alespoň ve třech za sebou generovaných vektorech. Výstup z se navrátí do nuly v tom okamžiku, kdy tato podmínka pomine. Řešte tak, že ve zvláštním kombinačním obvodu tvořeném hradly nonekvivalence vyhodnotíte paritu vstupního vektoru, výstupní jednobitovou veličinu využijte jako vstup navazujícího sekvenčního obvodu.



4) Navrhněte synchronní sekvenční obvod se vstupem $[x]$ a s výstupem $[z]$. Obvod porovnává bity vstupujícího slova. Výstup $z = 1$ pouze tehdy, když se na vstupu objeví posloupnost 1011 nebo 1001. Výstup z se navrátí do nuly zároveň s první další 0 na vstupu x .

5) Navrhněte synchronní sekvenční obvod se vstupy $[x_0, x_1, x_2]$ a s výstupem $[z]$. Obvod porovnává vstupní 3bitovou informaci s vektorem 011. Na výstupu z se v případě rovnosti vstupního a klíčového vektoru vygeneruje v následujících třech hodinových cyklech sériově posloupnost 011.

6) Navrhněte synchronní sekvenční obvod s třemi vstupy $[x_0, x_1, x_2]$ a s výstupem $[z]$. Obvod porovnává vstupní 3bitovou informaci s vektorem 101 a v následujícím taktu hodin s vektorem 011. Na výstupu z se v případě rovnosti vektorů nastaví hodnota 1. Výstup se navrátí do nuly zároveň s dalším taktem hodinového signálu.

7) Navrhněte 5 stavový čítač v Grayově kódu. Čítač opatřete synchronním resetem.

8) Navrhněte cyklický binární čítač pro čítání nahoru od čísla N do 3, kde N je hodnota předvolby.

9) Navrhněte tříbitový cyklický čítač pro čítání nahoru a dolů (vpřed a vzad). Čítač je řízen hodinovým signálem, signálem pro směr čítání a resetovacím synchronním signálem.

10) Navrhněte elektronický obvod, který vyhodnocuje směr otáčení. Ze snímače otáček vstupují do obvodu dva vzájemně posunuté signály se střídou 1:1 o stejné frekvenci. Podle jejich vzájemného posunutí (předbíhání či zpoždování) elektronický obvod na svém výstupu vyhodnocuje směr otáčení vpravo nebo vlevo (log. 0 nebo 1). Vyhodnocování stavu signálů (vzorkování) provádějte v časových intervalech nezávislých na vstupních signálech.